

МАГІСТЕРСЬКА РОБОТА

МР. ІІМ - 18.00.00.000 ІІЗ

Група ІІМ-22-1

Чучук Дана

2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Чучук Дана Віталіївна

(прізвище, ім'я, по батькові)

УДК 004.942
(індекс)

МАГІСТЕРСЬКА РОБОТА

Методи та алгоритми планування маршрутів з використанням нечітких

моделей

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Чучук Д.В.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник

Пасека Микола Степанович, д.т.н., професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

В.о. завідувача кафедри

доц.

Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Рецензент

доц.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

В.о. зав. кафедрою ІІЗ

доц. В.В. Бандура

“ 04 ” вересня 2023 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Чучук Дані Віталіївні

(прізвище, ім'я, по-батькові)

1. Тема магістерської роботи “ Методи та алгоритми планування маршрутів з використанням нечітких моделей ”

керівник проекту (роботи) Пасека Микола Степанович, д.т.н., професор

затвердені наказом закладу вищої освіти від “ 18 ” грудня 2023 р. № 738/7

2. Строк подання студентом проекту (роботи) 15 січня 2024 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних технологій оптимізації маршрутів

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Аналіз проблеми планування та оптимізації логістичних маршрутів

2. Алгоритмічне забезпечення інформаційної системи побудови логістичних маршрутів

3. Оптимізація логістичних маршрутів на основі використання імітаційних моделей

4. Прикладна реалізація системи побудови планування маршрутів на основі нечіткої логіки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Розв'язок TSP проблеми (рис. 1.1)

2. Варіанти конфліктних ситуацій (рис. 2.1)

3. Фазифікація якісних вхідних сигналів з ФН трикутної форми (рис. 2.2)

4. Узагальнений алгоритм фазифікації (рис. 2.3)

5. Шкала лінгвістичних термів (рис. 2.4)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Нормоконтроль	доц., к.т.н. Вовк Р.Б.	
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2023 р.

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	01.10.2023	виконано
2	Аналіз концепцій та алгоритмів предметної області	25.10.2023	виконано
3	Аналіз проблеми планування та оптимізації логістичних маршрутів	10.11.2023	виконано
4	Алгоритмічне забезпечення інформаційної системи побудови логістичних маршрутів	22.11.2023	виконано
5	Оптимізація планування маршрутів на основі використання імітаційних моделей	01.12.2023	виконано
6	Реалізація функціональності запропонованої інформаційної технології	15.12.2023	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.01.2024	виконано

Студент – магістр

_____ (підпис)

Керівник роботи

_____ (підпис)

АНОТАЦІЯ

Магістерська робота: 97 с., 24 рис., 3 табл., 45 джерел.

Тема: Методи та алгоритми планування маршрутів з використанням нечітких моделей

Об'єкт дослідження: інформаційна технологія планування логістичних маршрутів та траєкторій в умовах невизначеності.

Мета роботи: дослідження процесу підвищення ефективності процесів побудови логістичних маршрутів і траєкторій рухомих об'єктів шляхом використання нечітких моделей та методів.

Предмет дослідження: моделі та алгоритми, що забезпечують підвищення ефективності планування оптимізації логістичних маршрутів засобами нечіткої логіки.

Результати дослідження:

Представлені алгоритми, досліджені моделі та система підтримки прийняття рішень на основі нечіткої експертної системи можуть бути використані для оптимізації логістичних маршрутів широкого класу рухомих об'єктів.

Висновок:

Розроблено структуру інтелектуальної інформаційної системи, базу знань нечіткої експертної системи для прийняття рішень шляхом прогнозування рівня сатисфакції нечітких замовлень в вузлах-претендентах та процедуру вибору критичного значення рівня на основі мінімізації загальної довжини маршрутів.

НЕЧІТКА ЕКСПЕРТНА СИСТЕМА, ЛОГІСТИЧНИЙ МАРШРУТ, ПРОГНОЗУВАННЯ, СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, НЕЧІТКА ІНФОРМАЦІЯ, НЕВИЗНАЧЕНІСТЬ.

ABSTRACT

Master Thesis: 97 pp., 24 fig., 3 tab., 45 sources.

Thesis Subject: Route planning methods and algorithms using fuzzy models.

Object of research: information technology for planning logistics routes and trajectories under conditions of uncertainty.

Research goal: the study of the process of increasing the efficiency of the processes of building logistics routes and trajectories of moving objects by using fuzzy models and methods.

Subject of research: models and algorithms that ensure an increase in the efficiency of planning optimization of logistics routes by means of fuzzy logic.

The results:

The presented algorithms, researched models and a decision support system based on a fuzzy expert system can be used to optimize logistics routes of a wide class of moving objects.

Conclusion:

The structure of the intelligent information system, the knowledge base of the fuzzy expert system for decision-making by predicting the level of satisfaction of fuzzy orders in applicant nodes and the procedure for choosing the critical value of the level based on the minimization of the total length of the routes have been developed.

FUZZY EXPERT SYSTEM, LOGISTICS ROUTE, FORECASTING, DECISION SUPPORTING SYSTEM, FUZZY INFORMATION, UNCERTAINTY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	8
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ МАРШРУТІВ І ТРАЄКТОРІЙ.....	12
1.1 Методи планування та оптимізації логістичних маршрутів і траєкторій рухомих об’єктів управління	12
1.2 Опис об’єктів, що функціонують в умовах невизначеності та нечіткості інформації	23
1.3 Опис моделей планування поведінки рухомих об’єктів.....	32
Висновки до розділу	39
РОЗДІЛ 2. ФОРМАЛЬНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОБУДОВИ ЛОГІСТИЧНИХ МАРШРУТІВ І ТРАЄКТОРІЙ	41
2.1 Фазифікація сигналів в експертних системах підтримки прийняття рішень	41
2.2 Структура нечіткої експертної системи для прийняття рішень на основі нечітких даних	47
2.3 Алгоритми агрегації нечітких множин та нечітка база правил на основі лінгвістичних термів	52
2.4 Оптимізація логістичних маршрутів на основі використання імітаційних моделей.....	57
2.5 Забезпечення кращої визначеності інформації в системах на основі нечіткої логіки	61
2.6 Опис алгоритмів пошуку найближчих вузлів-претендентів	64
2.7. Моделі, що враховують часові та інші обмеження при плануванні логістичних маршрутів	67

	7
Висновки до розділу	71
РОЗДІЛ 3. ПРИКЛАДНА РЕАЛІЗАЦІЯ СИСТЕМИ ПОБУДОВИ	
ЛОГІСТИЧНИХ МАРШРУТІВ НА ОСНОВІ НЕЧІТКИХ МОДЕЛЕЙ.....	72
3.1 Основні концепції проекту системи підтримки прийняття рішень	72
3.2 Об'єктна декомпозиція задачі планування логістичних маршрутів та розробка діаграм класів	74
3.3 Опис об'єктної моделі	83
Висновки до розділу	86
ВИСНОВКИ.....	87
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	88
ДОДАТКИ.....	92

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

СППР – система підтримки прийняття рішень

НЕС – нечітка експертна система

НБЗ – нечітка база знань

ЕС – експертна система

БЗ – база знань

ЛПР - людина, яка приймає рішення

ВСТУП

Актуальність теми. Проблеми створення високоефективних систем підтримки прийняття рішень (СППР) для оптимізації процесів планування логістичних маршрутів та траєкторій суттєво впливають на ефективність функціонування рухомих об'єктів управління, зокрема на ефективність транспортних перевезень, бункерувальних операцій, а також на ефективність робототехнічних систем, маніпулятори яких відтворюють складні траєкторії в процесі виконання технологічних операцій, та ін. Задачі такого типу значно ускладнюються при їх розв'язанні в умовах невизначеності апріорної і поточної інформації. До цього класу відноситься і задача планування економічних маршрутів суден з точки зору забезпечення мінімізації їх подовженості, зменшення часової тривалості та зниження витрат палива на реалізацію маршруту. Її розв'язок часто ускладнюється високим ступенем невизначеності апріорної і поточної інформації в вузлах маршрутів та траєкторій, зокрема, при плануванні маршрутів танкерів, які мають доставляти паливо суднам, розміщеним в різних портах (вузлах дислокації). Замовлення портів на етапі попереднього планування маршрутів, як правило, є нечіткими, наприклад, "біля 500 т", "в середньому 300 т", "не менше 200 т" і т.п., тобто розмитою або нечіткою є інформація про обсяг вантажу, що має бути доставленим в кожний з вузлів на маршруті. Математична формалізація задач такого класу з врахуванням умов невизначеності апріорної інформації є досить складною проблемою. Для аналізу та оптимізації морських перевезень, взаємодії суден та розв'язання інших задач планування в морській логістиці найефективнішими, з точки зору математичної формалізації задач і шляхів їх розв'язання, є методи дослідження операцій, теорії нечітких множин, нечіткої логіки та теорії прийняття рішень.

На теперішній час зусиллями українських та іноземних вчених вже розв'язано багато задач, пов'язаних з плануванням маршрутів та траєкторій рухомих об'єктів, що функціонують в детермінованих або стохастичних

умовах. Разом з тим, ще недостатньо розроблені математичні моделі та алгоритми прийняття рішень для формування маршрутів і траєкторій в умовах невизначеності поточної і апріорної інформації, представленої якісними лінгвістичними моделями. Подальшого розвитку й удосконалення вимагає також відповідне програмно-алгоритмічне забезпечення СППР.

Отже, підвищення рівня автоматизації багатоетапних процесів прийняття рішень при управлінні складними, зокрема, рухомими об'єктами і системами в умовах невизначеності є одним з найбільш перспективних напрямків застосування, розробки і створення сучасних високоінтелектуальних технологій та програмного забезпечення.

Метою магістерської роботи є дослідження процесу підвищення ефективності процесів побудови логістичних маршрутів і траєкторій об'єктів в умовах невизначеності та ресурсних обмежень шляхом використання нечітких моделей.

Досягнення мети здійснюється розв'язанням таких **основних задач**:

- аналіз існуючих методів планування логістичних маршрутів і траєкторій в умовах невизначеності;

- синтез комбінованих алгоритмів оптимізації маршрутів та траєкторій в умовах невизначеності на основі нечітких моделей та нечітких експертних систем;

- синтез імітаційних моделей для дослідження та тестування СППР при плануванні логістичних маршрутів.

Об'єкт дослідження – інформаційна технологія планування логістичних маршрутів та траєкторій в умовах невизначеності.

Предмет дослідження: моделі та алгоритми, що забезпечують підвищення ефективності планування оптимізації логістичних маршрутів засобами нечіткої логіки.

Методи дослідження. Проведені в магістерській роботі дослідження базуються на комплексному використанні методів дослідження операцій; теорії нечітких множин; теорії нечіткої логіки; теорії випадкових процесів.

Наукова новизна одержаних результатів полягає в розвитку і поглибленні теоретичних основ формування нечітких моделей та алгоритмів для підвищення ефективності процесів формування логістичних маршрутів в умовах невизначеності і неповноті даних.

Практичне значення отриманих результатів полягає в тому, що представлені алгоритми, досліджені моделі та система підтримки прийняття рішень на основі нечіткої експертної системи можуть бути використані для оптимізації логістичних маршрутів широкого класу рухомих об'єктів.

Структура магістерської роботи. Робота складається з вступу, трьох розділів та переліку використаних джерел. Загальний обсяг магістерської роботи складає 97 сторінок і містить 24 рисунки, 3 таблиці та посилання на 45 літературних джерел.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛОГІСТИЧНИХ МАРШРУТІВ І ТРАЄКТОРІЙ

1.1 Методи планування та оптимізації логістичних маршрутів і траєкторій рухомих об'єктів управління

Задачі планування ефективних логістичних маршрутів та траєкторій суттєво впливають на показники ефективності транспортних перевезень в цілому. Це пов'язано, перш за все, з оптимізацією енергетичних та часових витрат при реалізації запланованих маршрутів та траєкторій. Часто задачі попереднього планування маршрутів та траєкторій розв'язують на основі відомих методів та алгоритмів дослідження операцій, в тому числі на основі методів лінійного та цілочисельного програмування. Досить широко використовуються підходи, що базуються на розв'язку класичної транспортної задачі та задачі комівояжера. Аналогічні проблеми виникають при плануванні маршрутів мобільних та підводних роботів, а також траєкторій маніпуляційних систем, що встановлені на рухомих основах. Складнощі в плануванні маршрутів та траєкторій виникають при необхідності врахування впливу зовнішнього середовища, що в загальному випадку може характеризуватись нестационарними збуреннями, а також при відсутності достатньо повної апріорної інформації про особливості реалізації запланованих маршрутів і траєкторій та існуючі обмеження в конкретних точках запланованих траєкторій. Особливо задача планування маршрутів та траєкторій ускладнюється при необхідності застосування моделей та алгоритмів для її розв'язання в умовах невизначеності.

Перш за все це пов'язано зі складністю формалізації реальних задач аналізу та планування морських перевезень як моделей дослідження операцій. Серед прикладів ефективного застосування методів дослідження операцій для синтезу алгоритмів прийняття рішень в умовах нестационарного морського середовища слід назвати: ресурсний менеджмент для торгового флоту,

планування розкладу ескадри суден, консенсус при прийнятті рішень, моделювання екстремальних умов, прийняття рішень в аварійних ситуаціях, класифікація аварійних ситуацій, аналіз поширення нафтових продуктів при морських аваріях. Особливе значення має розв'язок транспортних задач та задач планування маршрутів суден.

Процеси планування маршрутів характеризуються як задачі оптимізації різноманітних рухомих об'єктів, що функціонують в нестационарних умовах або в умовах невизначеності. Прикладне застосування теоретичних підходів та результатів доцільне з точки зору мінімізації сумарної довжини маршрутів при: плануванні замкнених траєкторій маніпуляційних роботів з багатьма точками позиціонування (як динамічних сцен) на різних етапах технологічних процесів; плануванні траєкторій мобільних роботів та підводних апаратів; плануванні автомобільних перевезень; плануванні маршрутів танкерів, що обслуговують судна-замовники, які розташовані в різних портах, та ін.

Задача планування маршрутів і траєкторій в різних постановках зводиться до розв'язання задач типу: задача комівояжера (TSP – Traveling Salesman Problem) та задача формування маршрутів (VRP – Vehicle Routing Problem).

В загальному вигляді проблему планування маршрутів можна сформулювати наступним чином. Існує множина вузлів (міст, портів, точок позиціонування), що мають в будь-якому порядку обслуговуватись, наприклад, групою агентів або парком машин (автомобілів, суден, роботів і т.п.). Маршрут – це послідовність вузлів, які агент повинен відвідати разом з зазначенням відповідних послуг, які він забезпечує. Необхідно синтезувати реальний набір економічних маршрутів для кожного агента у вигляді розв'язків просторової задачі без часових або інших обмежень, окрім обмежень на максимальну довжину маршруту.

На рис. 1.1. наведено приклад формування маршрутів, до складу яких входять 15 вузлів ($N + N_0$), включаючи два початкові (базові) вузли A та B ($N_0 = 2$). Кожен вузел має попит в 1 одиницю (наприклад, кількість необхідного для розвантаження вантажу), а можливість машини $D = 3$ одиниці (D –

вантажомісткість машини), при чому кожний маршрут повинен починатися і закінчуватися у відповідному базовому вузлі. Зв'язані графи, як математичні моделі маршрутів, можуть бути орієнтованими або неорієнтованими в залежності від врахування асиметрії або симетрії відстаней між вузлами, відповідно.

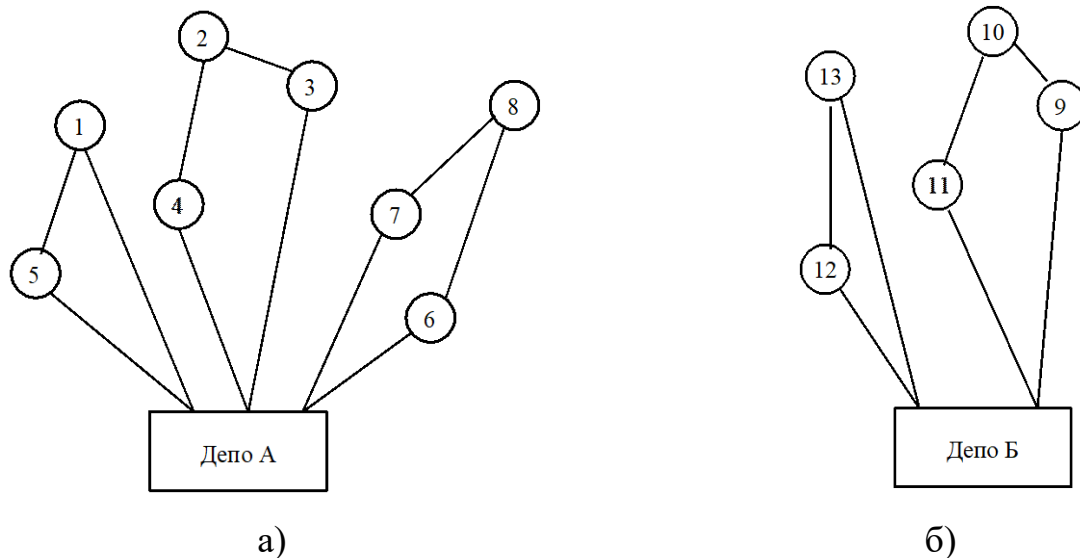


Рис. 1.1. Розв'язок TSP проблеми:

- а) - для базового вузла А - Маршрут R1: А – 1 – 5 – А;
 Маршрут R2: А – 3 – 2 – 4 – А; Маршрут R3: А – 6 – 7 – 8 – А;
 б) - для базового вузла В - Маршрут R4 В – 12 – 13 – В;
 Маршрут R5: В – 9 – 10 – 11 – В

Через складність наведених задач їх розв'язок комбінаторні аналітики досліджують у напрямках розв'язку: задачі комівояжера (*TSP* проблеми з одним агентом); *M-TSP* проблеми (з кількома агентами), коли будь-який з агентів має практично необмежені можливості і має відвідати хоча б один вузол при відсутності обмежень на загальну кількість вузлів на маршрутах де r – кількість маршрутів; класичної задачі вузлового планування маршрутів (*VRP*) з одним базовим вузлом $N_0=1$ та кількома агентами з обмеженими можливостями (приклад для $N=8$ та $D=3$ наведено на рис. 1.1, а); *VRP* проблеми з кількома базовими вузлами (рис. 1.1: $N=13$, $r=5$, $D=3$, $N_0=2$); *VRP* проблеми із

стохастичним характером замовлень (попиту) вузлів $q_j = \text{var}$, $j = 1 \dots N$, коли замовлення апріорно невідомі і формуються на основі того чи іншого стохастичного закону розподілу.

Аналізуючи в подальшому різні методи та алгоритми, за допомогою яких розв'язуються задачі планування маршрутів, відзначимо, що деякі з цих алгоритмів можуть бути представлені найнеоптимальнішими рішеннями. Для цих рішень найгірше значення цільової функції не перевищує в $K(N)$ разів її оптимального значення, де $K(N)$ – це функція від кількості вузлів N . Для оптимальних алгоритмів $K(N)=1$, а для евристичних алгоритмів, як правило, $K(N)>1$ і при цьому збільшується при зростанні кількості вузлів N . Після визначення найгіршого рішення, необхідно здійснювати зворотній математичний аналіз для пошуку оптимальних рішень.

Розглянемо більш детально математичні моделі, а також точні та евристичні алгоритми для розв'язку першого типу задач планування маршрутів (*TSP* проблеми з одним агентом).

Нехай $G = (V, A)$ є граф, де V – множина з N вершин (вузлів); A – множина дуг (ребер) графа, а $C = (c_{ij})$ є матриця відстаней між вузлами, що асоціюється з A . Розв'язок задачі полягає в визначенні маршруту з мінімальною довжиною, що проходить через кожен вузол графа тільки один раз. Такий маршрут називають Гамільтоновим ланцюгом або Гамільтоновим циклом. У випадку симетричної матриці C має місце співвідношення

$$\forall i, j \in V: c_{ij} = c_{ji},$$

а у випадку асиметричної матриці C - співвідношення $c_{ij} \neq c_{ji}$, $i, j \in V$. При цьому всі елементи матриці C відповідають умові трикутної нерівності

$$\forall i, j, k \in V: c_{ij} + c_{jk} \geq c_{ik}$$

Практично при розв'язку *TSP* проблеми необхідно з'ясувати чи має граф $G=(V, A)$ Гамільтонів цикл (HC). Розглянемо математичні моделі *TSP* проблеми в постановці задачі лінійного цілочисельного програмування. Проаналізуємо DFG-модель, що базується на логічних змінних x_{ij} для кожної дуги графа (i, j) , які тільки при умові входження (i, j) до оптимального рішення дорівнюють 1. При цьому DFG-модель, в якій S – підмножина вершин, має наступний вигляд:

- цільова функція:
$$\min \sum_{i \neq j} c_{ij} x_{ij} \quad (1.1)$$

- обмеження:
$$\left\{ \begin{array}{l} \sum_{j=1}^N x_{ij} = 1, \quad i = 1 \dots N, \\ \sum_{i=1}^N x_{ij} = 1, \quad j = 1 \dots N, \\ \sum_{i, j \in S} x_{ij} \leq |S| - 1, \\ S \subset V, 2 \leq |S| \leq N - 2, \\ x_{ij} \in [0, 1], \\ i, j = 1 \dots N, \quad i \neq j. \end{array} \right. \quad (1.2)$$

Наведена модель містить $N(N-1)$ логічних змінних, $2N$ обмежень, $(2^N - 2N - 2)$ обмежень, що забороняють формування підциклів, і навіть для середньої кількості вузлів N її реалізація вимагає досить значних обчислювальних ресурсів.

MTZ-моделі базуються на використанні додаткових змінних $u_i, u_j (i, j = 2 \dots N)$ для зменшення числа підциклових обмежень, які при цьому можна записати наступним чином:

$$\begin{aligned}
 u_i - u_j + (N - 1)x_{ij} &\leq N - 2 \\
 i, j &= 2 \dots N, \quad i \neq j, \\
 1 \leq u_i &\leq N - 1, \quad i = 2 \dots N
 \end{aligned}$$

Модифікована модель вищенаведених обмежень базується на додаткових компонентах, що вводяться в ліву частину нерівності

$$\begin{aligned}
 u_i - u_j + (N - 1)x_{ij} + (N - 3)x_{ij} &\leq N - 2 \\
 i, j &= 2 \dots N, \quad i \neq j.
 \end{aligned}$$

Реалізація MTZ-моделей вимагає значно менших обчислювальних ресурсів, але в деяких випадках DFG-модель дозволяє отримати кращі рішення.

Розглянемо в подальшому евристики трьох класів для виконання: процедур конструювання маршрутів, процедур покращення маршрутів та комбінованих процедур.

Процедура конструювання маршрутів генерує приблизно оптимальний шлях на основі матриці відстаней. Процедура покращення маршрутів намагається знайти найкращий шлях на основі початкового маршруту. Комбінована процедура об'єднує задачі двох попередніх процедур. Припустимо, що матриця C симетрична ($c_{ij} = c_{ji}$).

Серед відомих евристик для конструювання маршрутів слід назвати циклічний алгоритм (з порядком обчислень N^2) пошуку найближчого вузла, що включається потім до маршруту, для якого максимально можливе значення відношення довжини сформованого маршруту $L_{НС}$ до довжини оптимального маршруту L_{opt} складає

$$\frac{L_{НС}}{L_{opt}} \leq \frac{1}{2} [\ln(N)] + \frac{1}{2}.$$

Суть економічного Clark і Wright алгоритму полягає в наступному:

Крок 1. Вибір базового вузла 1.

Крок 2. Обчислення значень (економій) $s_{ij} = c_{1i} + c_{1j} - c_{ij}$ для $i, j = 2, 3, \dots$

N .

Крок 3. Ранжування значень s_{ij} від найбільшого до найменшого.

Крок 4. Починаючи з першого елемента ранжованого ряду і рухаючись до його кінця формується довший маршрут, шляхом з'єднання відповідних вузлів i та j . Процедура повторюється до повного формування НС.

При реалізації цього евристичного алгоритму на кроці 2 для обчислення матриці $S = [s_{ij}]$ необхідно виконання cN^2 операцій для деякої константи c . На кроці 3 значення s_{ij} можуть бути відсортовані шляхом використання методу пірамідального сортування при максимальному значенні $cN^2 \lg(N)$ порівнянь та переміщень. Крок 4 потребує виконання порядку N^2 операцій. Таким чином, Clark і Wright алгоритм потребує виконання операцій порядку $N^2 \lg(N)$.

Евристичний алгоритм вставки полягає в тому, що для сформованого за k ітерацій підшляху з k вузлів визначається вузол-претендент, який не належить цьому підшляху і повинен до нього під'єднуватися на наступному кроці процесу формування маршруту, а потім визначається місце в підшляху, де $(k+1)$ -й вузол має бути вставлений (крок вставки). Більш детально розглянемо декілька різновидностей алгоритму вставки, враховуючи, що при апріорно відомому базовому вузлі складність всієї процедури зменшується пропорційно значенню N .

Алгоритм найближчої (найвіддаленішої) вставки полягає в наступному:

Крок 1. Починається з підграфу, що містить тільки вузол i .

Крок 2. Визначається вузол-претендент k , для якого значення c_{ik} є мінімальним та формується підшлях $(i - k - i)$.

Крок 3. Крок вибору. Для даного підшляху визначається новий вузол k , що в нього не входить і є найближчим (найвіддаленішим) до будь-якого вузла цього підшляху.

Крок 4. Крок вставки. Визначається ребро (i, j) в підшляху, яке мінімізує значення $c_{ik} + c_{kj} - c_{ij}$. Вузол k вставляється до підшляху між вузлами i та j .

Крок 5. Перехід до кроку 3 доки не буде сформовано НС.

Для цього алгоритму $\frac{L_{НС}}{L_{opt}} \leq 2$, а кількість обчислень має порядок N^2 .

Використовуються також алгоритми найефективнішої, вільної, швидкої, випуклої, різницево-рейтингової вставки та вставки за найбільшим кутом.

На використанні мінімальних остових дерев базуються алгоритми *мінімального остового дерева* (Kim) та *евристики Christofides*, для яких,

відповідно, мають місце наступні співвідношення: $\frac{L_{НС}}{L_{opt}} \leq 2$, $\frac{L_{НС}}{L_{opt}} \leq 1,5$.

Алгоритм *найближчого сполучення* (Rosenkrantz, etc.) для розв'язку TSP проблеми з n вузлами формує послідовність S_1, \dots, S_N , в якій кожне S_i - це множина $(n - i + 1)$ окремих підшляхів, що охоплюють всі N вузлів. Суть алгоритму полягає в наступному:

Крок 1. S_1 складається з N підшляхів, кожен з яких містить тільки один вузол.

Крок 2. Для всіх $i < N$ знаходимо ребро (a_i, b_i) , для якого виконується умова $c_{a_i b_i} = \min(c_{xy})$ для вузлів x та y з різних підшляхів S_i . Потім S_{i+1} формується з S_i об'єднанням підшляхів, що містять a_i та b_i (на кожному кроці два найближчі підшляхи поєднуються один з одним).

Об'єднання двох підшляхів T_1 і T_2 , коли один з них або обидва містять тільки один вузол, є тривіальним. Якщо ж T_1 та T_2 містять хоча б по два вузли, то існують ребра (a, b) в T_1 та (d, e) в T_2 , для яких значення $(c_{ad} + c_{be} - c_{ab} - c_{de})$ є мінімальним. При об'єднанні (T_1, T_2) утворюється шлях, отриманий з T_1 та T_2 видаленням (a, b) та (d, e) і додаванням (a, d) та (b, e) .

Для наведеного алгоритму $\frac{L_{НС}}{L_{opt}} \leq 2$, а порядок кількості обчислень складає N^2 .

Можливо найбільш відомими евристиками для *покращення маршрутів (шляхів)* є евристики обміну гілок, зокрема евристики 2-opt і 3-opt, запропоновані Lin та евристика k -opt для $k \geq 3$ – Lin і Kernighan.

Алгоритм формування маршрутів на основі *евристик обміну гілок*:

Крок 1. Знаходиться початковий шлях. Звичайно цей шлях довільно обирається з множини всіх можливих шляхів.

Крок 2. Покращується шлях на основі однієї з евристик обміну гілок.

Крок 3. Перехід до кроку 2 доки не буде зроблено додаткове покращення.

Процедура обміну гілок завершується в локальному оптимумі. Для даного k , якщо визначається k -змін маршруту, новий маршрут формується шляхом видалення k гілок з поточного маршруту і заміною їх іншими k гілками. Маршрут вважається k -оптимальним, якщо його неможливо покращити шляхом k -змін. Кроки 2 та 3 повинні сформувати цей k -оптимальний маршрут. Чим більше значення k , тим кращого локального оптимуму вдається досягти.

Аналіз найгірших рішень показує, що в деяких випадках 2-opt маршрут може бути вдвічі довшим, ніж оптимальний маршрут. Разом з тим процедура 3-opt потребує в N разів більше часу, ніж процедура 2-opt. Щоб отримати найкраще рішення потрібно повторювати процедуру 3-opt для багатьох початкових варіантів. Щодо обчислень, при зростанні N цей метод швидко стає незручним.

Більшість з розглянутих вище евристичних алгоритмів можуть бути застосовані для розв'язку як симетричних, так і асиметричних TSP.

Пошук більш ефективних евристичних алгоритмів розв'язку TSP продовжується, зокрема, шляхом дослідження трансформації матриці відстаней і обмежень, на основі комбінованого алгоритму GENIUS, застосування субмодальних функцій та методів штучного інтелекту.

Для розв'язку проблем планування маршрутів в стандартній постановці задачі VRP (з врахуванням обмежень на можливості агентів) також використовуються точні та евристичні моделі і алгоритми.

Точні математичні моделі забезпечують мінімізацію загальної довжини неперервних замкнених маршрутів з задоволенням попиту в кожному вузлі

тільки одним агентом та з врахуванням загальних часових обмежень та синхронізації часових обмежень в вузлах (часові вікна).

Більшість стратегій рішень при плануванні маршрутів (VRP) можуть бути класифіковані наступним чином:

а) групування (кластеризація) – маршрутизація: спочатку згруповують вузли попиту, а на другому кроці розробляють економічні маршрути для кожної групи;

б) маршрутизація – маршрутизація: спочатку формується великий (звичайно неможливий для виконання) маршрут або цикл, що містить всі об'єкти попиту (вузли), а потім цей маршрут поділяється на менші з можливістю реалізації з урахуванням обмежень маршрути;

в) економічність / вставка: базується на порівнянні поточної конфігурації маршруту з альтернативною конфігурацією на основі відповідного критерію;

г) покращення / заміна: базується на покращенні маршрутів в напрямку їх оптимізації;

д) з використанням методів математичного програмування;

е) інтерактивна оптимізація: це алгоритм загального призначення, в якому людина, що приймає рішення (ЛПР), має великий вплив на процес, оскільки вона має можливість вводити в модель оптимізації змінні параметри та суб'єктивні оцінки на основі знань та інтуїції; це майже завжди збільшує ймовірність того, що кінцева модель реалізується та буде використовуватись;

є) точні процедури: спеціалізований метод гілок і меж, метод динамічного програмування, метод відтинаючих площин та ін.

Точні алгоритми оптимізації маршруту, що базуються на призначенні нижніх меж та використанні методу гілок та меж, використовують взаємозв'язок між VRP та M-TSP проблемами та математичні моделі на основі логічних змінних, що дозволяють формувати маршрути для різних постановок задач VRP, зокрема, для задачі CVRP з обмеженнями на ємність (вантажомісткість) агентів та задачі DVRP з обмеженнями на довжину будь-

якого маршруту або на час його реалізації. Оптимальні рішення отримано для CVRP з кількістю вузлів на всіх маршрутах – 260.

Модель симетричної VRP проблеми, яка сформована з використанням з k -ступеневого центрального дерева та обмежень у вигляді функції Лагранжа, дозволяє отримати оптимальні маршрути для діапазону вузлів 10...25. Аналогічні результати отримано при використанні метода динамічного програмування для розв'язку задачі CVRP, а в роботі наводяться оптимальні рішення для 50 вузлів. Широке застосування знайшли також моделі, що розроблені на основі сегментації та генерування колонок, використання яких в комбінації з іншими моделями призводить до формування оптимальних рішень в межах VRP з кількістю вузлів – до 100.

Особливої уваги заслуговують математичні моделі, що базуються на використанні 2-індексних, 3-індексних та багато-індексних змінних для представлення потоків машин або агентів в задачах VRP. На їх основі в роботі отримано оптимальні рішення (за відповідну кількість кроків обчислень) для VRP з кількістю вузлів – до 60. Оскільки точні алгоритми є “життєздатними” для задач з невеликою кількістю вузлів, то доцільно розглянути також основні евристичні алгоритми, що є ефективними для розв'язку VRP проблеми з більшою кількістю замовників (вузлів). Одним з найбільш поширених є економічний алгоритм Clark та Wright, основні аспекти якого на основі співвідношення $(c_{li} + c_{lj} - c_{ij})$ розглянуто раніше. Метод розгортки, що базується на використанні системи полярних координат, є ефективним для задач з розмірністю до 250 вузлів, дозволяє знайти розв'язок за два етапи. Спочатку він пов'язує вузли з машинами, а потім згідно послідовності замовлень кожна машина відвідує вузли, які з нею пов'язані. З початкового вузла розгортається (за або проти годинникової стрілки) промінь. Вузлі-претенденти додаються до маршруту в тій послідовності, в якій вони зустрічаються.

Двофазний алгоритм розроблено Christofides, Mingozzi та Toth на основі підходу з послідовно-паралельним формуванням маршрутів. Застосування

штрафних множників до цільових функцій у випадках перевищення попиту над можливостями покладено в основу гами штрафних алгоритмів (Stewart, Golden), розроблених для розв'язку VRP проблеми.

В останні роки досить ефективно застосовуються метаевристичні алгоритми табу-пошуку рішень, що базуються на формуванні послідовності рішень з їх наступним покращенням і направляють локальну процедуру евристичного пошуку на розширення простору рішень за межами області локального оптимуму. Результати моделювання показують, що ці алгоритми є одними з найкращих для розв'язку VRP проблем. Крім того, в багатьох дослідженнях останніх років для розв'язку VRP проблем широко застосовуються алгоритми, що базуються на методах штучного інтелекту. Серед них в першу чергу слід назвати: генетичні алгоритми, які маніпулюють двійковими рядками – хромосомами і моделюють еволюційні процеси; алгоритми моделювання обпалювання (simulated annealing), стратегія яких встановлює аналогію між проблемами розв'язку складних комбінаторних задач і теоретичними результатами в області термодинаміки; нейронні мережі, що базуються на імітації біологічних нейронів та їх з'єднань.

1.2 Опис об'єктів, що функціонують в умовах невизначеності та нечіткості інформації

Проблеми оптимізації процесів планування логістичних маршрутів чи траєкторій серйозно впливають не тільки на ефективність транспортних перевезень, а й на ефективність широкого класу багатоканальних робототехнічних систем, маніпулятори яких відтворюють складні траєкторії в процесі виконання операцій з об'єктами маніпулювання. Задачі такого типу значно ускладнюються при їх розв'язанні в умовах невизначеності.

Особлива увага серед найважливіших задач в області робототехніки приділяється підвищенню ефективності керованої поведінки маніпуляційних систем (МС), що знаходяться під впливом як зовнішніх збурень, так і впливом

збурень, що створюються рухомою основою. Останній фактор суттєво ускладнює задачу формування і реалізації бажаних траєкторій.

Проблема забезпечення бажаних показників якості та технічних характеристик (швидкодія, точність відтворення траєкторії, маневреність, енергоспоживання та ін.) при проектуванні керованих МС пов'язана з оптимальним вибором кінематичної структури МС, типу приводів, сенсорної системи, системи управління, стратегічний рівень якої має забезпечувати формування ефективних траєкторій в умовах існуючих збурень зовнішнього середовища. На точність відтворення бажаних траєкторій суттєво впливає також елементна база систем управління.

При плануванні траєкторій необхідно також враховувати, що при взаємодії робота з навколишнім середовищем виникають також механічні неточності гравітаційного та вібраційного походження, що породжуються коріолісовим і відцентровим прискореннями при русі МС, а також різними видами деформацій, в т.ч. пружними і пластичними. При використанні спеціальних програмних пристроїв та ЕОМ для розрахунку бажаних програмних траєкторій точність їх реалізації також залежить від похибок методів апроксимації, лінеаризації, помилок округлення і т.п. На основі статистичного аналізу похибок робота і розмірних похибок можна завжди визначити кількість можливих зіткнень хвата з об'єктами маніпулювання, що робить неефективним виконання, наприклад, складальних операцій. При цьому, в більшості випадків, статистичний розподіл розв'язувальної спроможності і похибки позиціонування приймають у вигляді гаусових (нормальних) розподілів.

Складність забезпечення бажаних технічних характеристик для керованих МС з рухомими основами пов'язана з поточними зміщенням основ, що призводить до ускладнення процесів формування та відтворення запланованих програмних траєкторій. При використанні побудованих на традиційних принципах автоматизованих (АСУ) та автоматичних систем

управління поставлені перед роботами задачі (через рухомість основ МС) в ряді випадків можуть залишитись невиконаними.

До класу МС с рухомою основою доцільно віднести:

- МС с закріпленою на судні-носії основою, що призначена для виконання спуско-підйомних операцій в умовах морського хвилювання різної бальності;
- МС підводних роботів (ПР), при функціонуванні яких можливе зміщення корпуса робота – основи МС, що може виникати через вплив на ПР течій, хвилювання та інших збурюючих впливів. Навіть тонкі кабелі, зокрема, оптичні волокна, формують для телекерованих ПР сильні змінні збурення;
- МС космічних станцій для одиночного або групового виконання складальних операцій, операцій монтажу і демонтажу, ремонту та заміни пошкоджених модулів і т.п.;
- МС мобільних робототехнічних систем на колісному або гусеничному ході для виконання технологічних операцій в будівництві, вугільній промисловості.

Вибір методів формування траєкторій МС з рухомою основою-платформою суттєво залежить від інерційних властивостей самих платформ.

При формуванні траєкторій МС підводних роботів, необхідно враховувати високий тиск робочого середовища. Клас задач ПР (огляд, зварювання, різання, випробування, зачистка, покриття, складання підводних конструкцій і т.п.) також впливає на вибір стратегії планування траєкторій, алгоритмічну організацію і структуру їх систем управління. Результати моделювання режимів роботи МС ПР вказують на необхідність розробки нової стратегії планування траєкторій, що дозволить звести до мінімуму вплив опору води і максимально ефективно використати підйомну силу. При цьому необхідно враховувати суттєві відмінності в динамічній поведінці МС в воді і в повітрі, оскільки у випадках низьких швидкостей траєкторних переміщень хвата МС приводи шарнірів з вертикальною віссю обертання вимагають більш високого рушійного моменту, а приводи вертикальних рухів – менших зусиль в

воді, ніж в повітрі. Отже, для МС ПР неможливо використання апробованих алгоритмів промислових роботів, а стратегія формування траєкторій МС ПР повинна враховувати динаміку ПР, взаємодію між МС і корпусом ПР, між ПР і морським середовищем.

Методи формування і реалізації траєкторій МС з рухомою основою можуть також застосовуватись для широкого класу систем точного наведення і спеціальних платформ, включаючи турельні приводи і великі антенні системи. Використання "моделі умовного робота" як 2-3-ступеневу МС з рухомою основою, методів спеціальної декомпозиції матриці інерції робота, методів компенсації нелінійностей і стратегії подвійного ПД-управління дозволяє досягти ефективної реалізації заданої траєкторії при швидкій збіжності алгоритмів в умовах впливу різних збурень.

При побудові багатоканальних систем управління програмним рухом в зварювальних та газорізальних машинах, промислових роботах, маніпуляторах, станках з програмним управлінням, радіотелескопах, вимірювальних та випробувальних стендах також необхідно розв'язувати серйозні проблеми ефективного формування і реалізації заданих траєкторій. Траєкторна похибка в таких системах залежить як від геометричних характеристик заданих траєкторій, так і від їх можливих "деформацій" в процесі реалізації, а її суттєве зменшення може бути досягнуто на основі визначення найкоротшої відстані між фактичною і заданою траєкторіями.

Розробка ефективних математичних моделей, що враховують особливості технологічних процесів та обмеження на процеси оптимізації і ресурси управління, дозволяє значно поліпшити програмно-алгоритмічну та апаратну реалізацію СППР, зокрема для автоматизації етапів інформаційної підготовки процесів прийняття рішень, формування альтернативних гіпотез, безпосереднього прийняття рішень та оцінки ефективності після їх реалізації. Так, при функціонуванні мехатронних, зокрема робототехнічних систем, в умовах динамічного середовища часто виникає необхідність в переміщенні хвата (захвата) МС в послідовний ряд фіксованих точок (позицій) робочого

простору з виконанням в кожній точці відповідної технологічної операції. При цьому траєкторія руху хвата між всіма точками робочого простору має виключати замкнуті підцикли (переміщення хвата двічі або більше разів в одну і ту ж саму точку), а розв'язання задачі оптимізації процесу планування траєкторії має базуватись на мінімізації загальної довжини сформованої траєкторії

$$\text{Min } L[\bar{\varphi}(i), \bar{\varphi}(j), n] = \text{Min } \sum_{i=0}^{n-1} \sum_{j=i+1}^n S[\bar{\varphi}(i) - \bar{\varphi}(j)],$$

де $\bar{\varphi}(i) = \{\varphi_x(i), \varphi_y(i), \varphi_z(i)\}$ – вектор координат i -ї точки робочої зони робота, з якої необхідно переміщувати хват на черговому кроці;

$\bar{\varphi}(j) = \{\varphi_x(j), \varphi_y(j), \varphi_z(j)\}$ – вектор координат j -ї точки робочої зони робота, в яку необхідно переміщувати хват на черговому кроці;

$S[\bar{\varphi}(i) - \bar{\varphi}(j)]$ – відстань між точкою i та точкою j траєкторії, що планується для переміщення хвата робота;

$L[\bar{\varphi}(i), \bar{\varphi}(j), n]$ – сумарна (загальна) довжина траєкторії;

n – кількість точок (вузлів) робочої зони робота, які мають знаходитись на траєкторії; $i, j = 0, 1, \dots, n; \quad i \neq j$.

В загальному випадку процес планування оптимальної траєкторії хвата робота являє собою процес оптимізації на основі просторової математичної моделі робототехнічної системи. Планетарна модель є частковим випадком просторової моделі, для якої

$$\forall \bar{\varphi}(i), \bar{\varphi}(j) \in R : \varphi_z(i) = 0, \varphi_z(j) = 0, \quad i, j = \overline{0, n}; \quad i \neq j,$$

де R – множина координат вузлів, що входять до робочої зони робота.

Прикладом застосування планетарних моделей може бути процес планування траєкторій руху мобільних (рухомих) роботів, що переміщуються

при виконанні технологічних операцій в горизонтальній площині. Для таких мобільних роботів функції $\bar{\varphi}(i)$, $\bar{\varphi}(j)$ мають вигляд

$$\bar{\varphi}(i) = \{\varphi_x(i), \varphi_y(i)\}, \quad \bar{\varphi}(j) = \{\varphi_x(j), \varphi_y(j)\}.$$

Слід зазначити, що в загальному випадку набір точок траєкторії, в які має бути переміщений хват робота, може змінюватись протягом всього технологічного процесу (збірка складних об'єктів, фарбування поверхонь складних деталей, транспортування та розподіл харчових компонентів та інші робототехнічні задачі в динамічних середовищах, в т.ч. для МС з рухомою основою та ін.).

Аналогічний підхід доцільно застосовувати до побудови планетарної моделі при плануванні оптимальних маршрутів танкерів для здійснення бункерувальних операцій. При цьому (в рамках одного маршруту) танкер має виходити з базового порту (початок маршруту), де знаходяться сховища з запасами паливних матеріалів, що необхідні для забезпечення розміщених в різних портах суден-замовників, і повертатися до базового порту (кінець маршруту).

Задача планування полягає в формуванні такого замкненого маршруту

$$\{\bar{\varphi}(0), \bar{\varphi}(1), \bar{\varphi}(2), \dots, \bar{\varphi}(n)\},$$

що проходить через кожний порт або кожну точку морської акваторії, в яких знаходяться судна-замовники, тільки один раз і при цьому також, як і при плануванні економічних траєкторій робота, забезпечується мінімізація загальної довжини маршруту.

Для вищенаведених задач планування маршрутів та траєкторії при чіткій апріорній інформації про кількість n і координати $\bar{\varphi}(i)$, $i = \overline{1, n}$ точок траєкторії або маршруту доцільно використовувати методи

цілочисельного програмування, зокрема метод гілок та меж по аналогії з його застосуванням для розв'язання задачі комівояжера, метод, що базується на використанні альтернативних змінних та інші методи дослідження операцій, що застосовуються для розв'язання задач типу VRP та TSP.

Практично проблеми планування траєкторії хвата в даній постановці задачі та аналогічні з точки зору математичної формалізації проблеми планування маршрутів мобільних роботів або танкерів-заправників базуються на множині вузлів, які мають бути обслуговані робототехнічною системою або танкером, відповідно. При цьому маршрут чи траєкторія являють собою послідовність станів, які хват, мобільний робот чи танкер мають відвідати разом із зазначенням (чи виконанням) відповідних технологічних функцій (завдань) в кожному з вузлів. Якщо танкер, наприклад, розвантажує суднове паливо (мазут, дизельне паливо) в кожному з портів, що знаходяться на траєкторії, і має кожного разу (після відпрацювання маршруту) повертатись в початковий порт для повного завантаження новим паливом, то математична модель такої проблеми може бути сформульована на основі додаткових обмежень, що входять до математичної моделі, сформульованої як модель CVRP (Capacitative Vehicle Routing Problem). До класу задач для об'єктів, планування і оптимізація маршрутів яких визначає ефективність їх функціонування, відносяться також задачі розробки загальної структури розподілених маршрутів для перевезення та постачання (шкільний або сільський автобус та ін.), задачі інформаційного забезпечення прийняття рішень при проектуванні параметрів маршрутів міського пасажирського транспорту, задачі планування маршрутів банківських посильних, де команда посильних отримує депозити в галузевих банках та повертає їх в центральний офіс для обробки, задачі планування маршрутів аварійних бригад в умовах зараженої місцевості (наявність завалів та пожегів, вплив газів, аерозолів, радіонуклідів, та інших небезпечних для здоров'я людей речовин та випромінювань, випадковість зовнішніх факторів: погодних умов, рози вітрів, ландшафту та ін.) при техногенних катастрофах та ін.

Окрім практичного значення маршрутизація та планування траєкторій є важливою проблемою з точки зору теоретичних доробок. Останні дослідження в цій області мають значні результати в проблемних формулюваннях і синтезі алгоритмів, що важливо також для пов'язаних з маршрутизацією та плануванням траєкторій досліджень в області комбінаторики та оптимізації обчислювальних процесів. Так методи розв'язання TSP та VRP проблеми можуть ефективно застосовуватись для широкого кола задач, зокрема, при монтажу комп'ютерних мереж і систем (трасування електропроводки комп'ютерних мереж і систем, при якій комп'ютерні модулі мають з'єднуватись між собою двома провідними трасами з мінімізацією їх загальної довжини), в паперовій промисловості (різання рулонних шпалер на листи з мінімізацією відходів), в металообробці (процеси штампування з мінімізованою за часом послідовністю операцій), планування різнотипних операцій на промисловому обладнанні, при моделюванні ігрових задач, в кристалографії (для оптимізації послідовності) експериментальних досліджень.

Перспективним для розв'язку задач попереднього планування маршрутів та в масштабі реального часу представляється використання методів штучного інтелекту, зокрема, нейронних мереж, методів на основі апріорно сформованих баз знань, методів фазі (нечіткої) логіки, використання генетичних алгоритмів та ін., що дозволяють ефективно розв'язувати задачі аналізу процесів обробки сенсорної інформації в реальному часі, побудови моделей зовнішнього середовища, навігації, формування траєкторій руху та обходу перешкод.

При використанні нечітких баз даних і нечітких систем знань, що базуються на знаннях у вигляді лінгвістичних стверджень групи експертів необхідно розробляти такі моделі і алгоритми, що будуть дозволяти доповнювати, виключати або модернізувати свою власну базу правил.

Безумовно, невизначеність апріорної або поточної інформації суттєво ускладнює розв'язок задач планування маршрутів і траєкторій, приводячи в деяких випадках до необхідності розв'язання конфліктних ситуацій в реальному масштабі часу. При цьому під конфліктною ситуацією будемо

розуміти таку ситуацію, при якій в процесі планування маршрутів і траєкторій виникає розумовне розходження між фактичними і бажаними значеннями окремих ресурсних параметрів, що суттєво впливають на ефективність процесів планування маршрутів і траєкторій, і пов'язані з необхідністю вибору ЛПР конкретного альтернативного рішення при наявності поточної інформації, критеріїв, вирішуючих правил, власної системи переваг та ін.

Основні фактори, що породжують конфліктні ситуації в АСУ реального часу (інтерактивні режими), можна класифікувати наступним чином:

- низька надійність елементів систем, що можуть виходити з ладу;
- недосконалістю самого процесу формування маршрутів та траєкторій, що обумовлюється неповнотою і неточністю інформації про об'єкт управління, недосконалістю моделей та алгоритмів, недоліками та помилками ЛПР;
- ресурсні обмеження (дистанція, вантажомісткість агентів, час та ін.);
- багатозначність при прийнятті рішень в реальних умовах;
- нездатність системи розв'язувати нові типи задач, що можуть виникати в умовах невизначеності апріорної чи поточної інформації.

В процесі планування маршрутів і траєкторій в реальних, динамічно змінюючихся умовах, наприклад, в умовах нестационарного морського середовища, проявляються всі основні джерела невизначеності інформації: невідомість, неповнота, недостовірність, випадковість, неточність та багатозначність.

Суднову транспортну задачу (СТЗ, STP) можна сформулювати як класичну транспортну задачу, яка може бути розв'язана методами лінійного програмування (ЛП) з відповідними модифікаціями стосовно реальних умов морського середовища. В загальній постановці СТЗ мінімізується цільова функція, що базується на оцінці ефективності транспортування різних видів вантажу (масло, вугілля, цемент, паливо та ін.) з декількох (одного) базових портів до різноманітних портів-споживачів (вузлів). Дуже важливим при розв'язанні СТЗ є врахування динамічного характеру морських умов, що приводить її до класу задач в нечіткому середовищі. Нечіткі моделі та

алгоритми на нечіткій логіці мають багато успішних застосувань, включаючи управління суднами та розв'язок СТЗ, зокрема: проблеми транспортування суднами вапна і цементу при використанні нечітких моделей лінійного програмування; планування морських перевезень вугілля з Тайпуру на основі нечітких двокритеріальних багатоіндексних моделей СТЗ; прийняття рішень при плануванні процесів транспортування гравію з використанням барж та відповідних вантажних операцій на основі нечітких моделей та алгоритмів на нечіткій логіці; планування маршрутів при взаємодії груп суден на основі теорії нечітких множин; багатокритеріальні СТЗ та ін. на основі нечітких моделей і алгоритмів. Особлива увага приділяється проблемі планування маршрутів та траєкторій в умовах стохастичних або нечітких замовлень портів-споживачів та нечітких значень відстаней між портами в умовах динамічного морського середовища.

1.3 Опис моделей планування поведінки рухомих об'єктів

При формуванні математичних моделей для оптимізації маршрутів і траєкторій суден, в т.ч. цільових функцій, в задачах математичного програмування необхідно враховувати особливості морського середовища, як джерела основних збурювальних впливів. Добре відомо, що морське середовище може створювати високодинамічні екстремальні умови і ступінь нечіткості (розмитості) параметрів та характеристик такого середовища є досить високою. Задачі планування маршрутів і траєкторій в умовах невизначеності найчастіше зводяться до задач CVRP та SVRP (Stochastic VRP). Задачі планування траєкторій маніпуляційних і мобільних роботів, що виконують технологічні операції в окремих точках робочого простору при ресурсних обмеженнях та нечіткій або ймовірнісній інформації про витрати відповідних матеріалів, наприклад, абразиву або води – для роботів, які автоматизують процеси очищення поверхонь, наприклад, корпусів суден, фарбових матеріалів – для роботів-фарбувальників і т.п., а також підводних

роботів, що збирають на значних глибинах конкреції з різними масо-габаритними характеристиками при обмеженнях на ємність вантажного модуля, є аналогами задач оптимального планування маршрутів танкерів, що здійснюють бункерувальні операції. Останнім задачам в їх узагальненій постановці доцільно приділити особливу увагу при обмеженнях, стосовно яких судна-замовники знаходяться в різних портах, а i -й танкер має заходити в j -й порт тільки один раз. Це пов'язано з часовими обмеженнями на проведення бункерувальних операцій, додатковими витратами власного палива при повторних заходах танкерів в порти, необхідністю повторного встановлення бонових загород в зоні взаємодії танкера з судном-замовником та додаткового супроводження бункерувального процесу пожежним судном: для дотримання портових вимог з екологічної (попередження розповсюдження нафтопродуктів на морській поверхні в аварійних ситуаціях) та пожежної безпеки.

Якщо танкер, наприклад, розвантажує суднове паливо (мазут, дизельне паливо) в кожному з портів, що знаходяться на траєкторії, і має кожного разу (після відпрацювання маршруту) повертатись в початковий порт для повного завантаження новим паливом, то математична модель такої проблеми може бути сформульована на основі додаткових обмежень, що входять до математичної моделі, сформульованої як модель CVRP:

а) цільова функція

$$J(x) = \sum_{i=0}^n \sum_{j=0}^n S[\bar{\varphi}(i), \bar{\varphi}(j)] x_{ij} \quad (1.3)$$

б) обмеження

$$\sum_{j=0}^n x_{ij} = 1, \quad i = \overline{1, n}; \quad (1.4)$$

$$\sum_{i=0}^n x_{ij} = 1, \quad j = \overline{1, n}; \quad (1.5)$$

$$\forall G \subseteq \bar{V} - \{0\}: \sum_{i \in N} \sum_{j \in N} x_{ij} < |G| - q(G), \quad |G| \geq 2; \quad (1.6)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = \overline{1, n}; \quad (1.7)$$

$$x_{ij} = \begin{cases} 1, & \text{if } S_{ij} \subset Route \\ 0, & \text{if } S_{ij} \not\subset Route \end{cases} \quad (1.8)$$

$$q(G) = \frac{\left(\sum_{i \in N} d_i \right)}{D}, \quad (1.9)$$

де $\bar{V} = \{V_0, V_1, V_2, \dots, V_n\}$ - множина вузлів маршруту, що, наприклад, представляють порти, в які має зайти танкер для обслуговування суден-замовників; $N = \{i_1, i_2, i_3, \dots, i_N\}$ - множина індексів i , що входять до комплектації поточного маршруту *Route*; G - поточна кількість вузлів, що входять до поточного маршруту в процесі його планування, на кожному кроці планування маршруту (введення додаткових вузлів до маршруту) цей параметр збільшується; S_{ij} - ребро або дуга графа, що характеризує відстань між вузлами i та j плануємого маршруту (траєкторії); $q(G)$ - величина, що характеризує ступінь розвантаження танкера при послідовному відвідуванні портів на маршруті, D - вантажомісткість танкера; d_i - величина замовлення i -го судна-замовника, тобто кількість судового палива, яке танкер-заправник має передати судну-замовнику.

Для наведеної математичної моделі (1.3) – (1.9) вважається, що замовлення будь-якого порту $d_i \leq D$ та сума всіх замовлень на будь-якому маршруті $\sum_{i \in N} d_i \leq D$ не перевищує повної вантажомісткості танкера D , а

кількісні характеристики замовлень $d_i (i=1...N)$ в кожному порту є апріорно відомі.

Якщо ж характер замовлень $d_i (i=1...N)$ кількості палива для суден-замовників, що розміщені на маршруті в портах, є стохастичним, то математична модель може бути сформована на основі стохастичних моделей, але при цьому необхідна інформація про закони розподілу густини ймовірностей замовлень як стохастичних параметрів.

Добре відомо, що апріорне отримання такої інформації дуже ускладнене, а інколи навіть неможливе. При цьому математична модель (1.3) – (1.9) має бути трансформована до SVRP-моделі. Оскільки при цьому в обмеженнях та цільовій функції SVRP-моделі за рахунок ймовірнісного характеру замовлень з'являються нелінійності, то це робить задачу SVRP значно складнішою ніж класична задача VRP. З цього приводу, для розв'язання SVRP-проблеми застосовуються тільки евристичні методи.

Деякі методи базуються на перетворенні SVRP-моделі в VRP-модель з подальшим застосуванням одного з існуючих алгоритмів розв'язання VRP. При цьому замість реальної вантажомісткості в алгоритмі VRP використовується штучна. Ця штучна вантажомісткість має менше значення ніж реальна. Маршрутизація здійснюється з формуванням середніх значень замовлень у вузлах, як штучних замовлень, при обмеженнях на вантажомісткість - у вигляді штучної вантажомісткості.

Якщо замовлення кожного порту є випадковою величиною, то задача планування маршрутів стає стохастичною задачею маршрутизації і має бути відповідно перетворена. У наведеній нижче математичній моделі замовлення d_i – випадкова величина з відомим стохастичним розподілом. Параметр α - це максимальна допустима ймовірність, для якої маршрут зазнає невдачі, тобто повністю не реалізується.

Стохастична модель маршрутизації танкерів (SVRP):

- цільова функція:
$$\text{Minimize } \sum_k \sum_{i,j} c_{ij} x_{ijk} \quad (1.10)$$

- обмеження:
$$\text{Prob} \left\{ \sum_{i,j} d_{ij} x_{ijk} \leq D \right\} \geq 1-\alpha, \quad k=1, \dots, r \quad (1.11)$$

$$x = [x_{ijk}] \in S, \quad (1.12)$$

де c_{ij} – дистанція між вузлами i та j ; D –вантажомісткість танкера; r – кількість танкерів; S – множина всіх рішень М-TSP задачі; $x_{ijk} = 1$, якщо вузли i та j вводяться до k -го маршруту або $x_{ijk} = 0$, якщо не входять. Обмеження (1.11) визначає шанс успішного рішення. Застосування наведеної моделі з ймовірнісним обмеженням доцільне, якщо штраф на порушення не може бути явно визначений. Головною в цьому випадку є суб'єктивна оцінка ЛПР з точки зору ймовірності невдачі маршруту. Ймовірнісне обмеження (1.11) може перетворюватися на еквівалентне детерміноване обмеження. В цьому випадку закрита форма виразу для штучної вантажомісткості може бути похідною від еквівалентного обмеження.

Розглянемо обмеження:

$$\text{Prob} \left\{ \sum_i a_i y_i \leq b \right\} \geq 1-\alpha \quad (1.13)$$

Якщо a_i є незалежною і рівномірно розподіленою випадковою величиною із середнім значенням \bar{a}_i і середньоквадратичним відхиленням s_i , а y_i - змінні рішень, то середнім значенням $\sum_i a_i y_i$ буде математичне сподівання

$M = \sum_i \bar{a}_i y_i$, а середньоквадратичним відхиленням $S = \sqrt{\sum_i s_i^2 y_i^2}$. Необхідно

знайти константу T , що задовольняє умові:

$$\text{Prob} \left\{ \frac{\sum_i a_i y_i - M}{S} \leq T \right\} = 1 - \alpha \quad (1.14)$$

Якщо $b \geq M + TS$, тоді ліва частина нерівності (1.13) також не менше, ніж $1 - \alpha$. Тому ймовірнісне обмеження (1.13) може бути замінене детермінованим обмеженням

$$M + TS \leq b \quad (1.15)$$

Це обмеження за походженням є нелінійним, оскільки його можна представити у формі:

$$\sum_i \bar{a}_i y_i + T \sqrt{s_i^2 y_i^2} \leq b \quad (1.16)$$

При спеціальних умовах обмеження (1.16) може набути лінійної форми. Якщо стохастичний розподіл є таким, що $S^2 = \lambda M$ для деякої константи λ (справедливо для гама, біноміальних та Пуасонівських розподілів), то (1.16) може бути приведено до вигляду

$$\sum_i \bar{a}_i y_i + T \sqrt{\lambda \sum_i \bar{a}_i y_i} \leq b \quad (1.17)$$

Або

$$M + T \sqrt{\lambda M} \leq b \quad (1.18)$$

Нерівність (1.18) трансформується до простої форми:

$$M \leq \bar{b} \quad (1.19)$$

Позначивши $w = T \lambda^{1/2}$ можемо записати

$$w M^{1/2} \leq b - M$$

і відповідно

$$M \leq (2b + w^2 - (w^4 + 4bw^2)) / 2 = \bar{b}. \quad (1.20)$$

На жаль, вираз \bar{b} (штучна вантажомісткість) за формулою (1.20) не існує для багатьох стохастичних розподілів. Крім того процедура отримання інформації відносно законів розподілу стохастичних замовлень дуже складна, а в багатьох випадках такої інформації просто не існує.

В реальних ситуаціях планування маршрутів в умовах невизначеності при формуванні і корекції математичних моделей необхідно враховувати:

- неточність оцінок або обчислення значень цільових функцій через неповноту знань про відстані між вузлами, що можуть бути визначені тільки у вигляді інтервалів. Оцінки $f(n)$ обчислюються за методи інтервального аналізу, що дозволяють знаходити найкращі рішення шляхом визначення і оцінки нижніх $\underline{f}(n_1)$ і верхніх $\bar{f}(n_1)$ границь неточних оцінок дистанцій (вартості) $[\underline{c}_{ij}, \bar{c}_{ij}]$ на протязі всього маршруту з використанням 4 можливих критеріїв класифікації $C1 \dots C4$, що або задовольняються або не задовольняються стосовно двох різних станів (дуг) n_1, n_2 :

$$\{C1, C2, C3, C4\} = \{ \underline{f}(n_2) \geq \bar{f}(n_1), \underline{f}(n_2) \geq \underline{f}(n_1), \bar{f}(n_2) \geq \bar{f}(n_1), \bar{f}(n_2) \geq \underline{f}(n_1) \};$$

- нечіткість оцінок значень цільових функцій через представлення параметрів цільової функції у вигляді нечітких інтервалів вимагає застосування методів теорії можливостей, що дозволяє визначати ступінь задоволення (сатисфакції) кожного з критеріїв $C1 \dots C4$ на основі понять необхідності Nec та можливості Pos : $C1$ оцінюється значенням $Nec(\underline{f}(n_2) > \bar{f}(n_1))$, $C2$ оцінюється

значенням $Nec \left(\underline{f}(n_2) \geq \underline{f}(n_1) \right)$, $C3$ оцінюється значенням $Pos \left(\bar{f}(n_2) \geq \bar{f}(n_1) \right)$, $C4$ оцінюється значенням $Pos \left(\bar{f}(n_2) \geq \underline{f}(n_1) \right)$;

- нечіткість координат вузлів маршрутів і траєкторій, для формалізації яких необхідне застосування методів теорії нечітких множин [1,89,136] з використанням поняття та значень “ступеню достовірності” та ін.;

- нечіткість замовлень (fuzzy demands) в вузлах маршрутів і траєкторій, формалізація яких вимагає застосування методів теорії нечітких множин а для прийняття рішень щодо включення вузла-претендента до поточної траєкторії необхідне застосування нечіткої логіки і нечітких експертних систем з використанням поняття та значень “рівня сатисфакції”, “ступеню привабливості” та ін.

Необхідність оцінки ефективності тих, чи інших алгоритмів для формування оптимальних траєкторій в умовах невизначеності вимагає розробки відповідних імітаційних моделей, що дозволяють моделювати процес планування і реалізації маршрутів та траєкторій в умовах різної інтенсивності зовнішніх збурень та наявності декількох джерел невизначеності апріорної та поточної інформації. Розробка таких імітаційних моделей дозволяє здійснювати апробацію різних алгоритмів планування маршрутів і траєкторій та приймати ефективні рішення стосовно вибору найкращих серед них з подальшим дослідженням їх поведінки в умовах нестационарних збурень та в умовах невизначеності.

Висновки до розділу

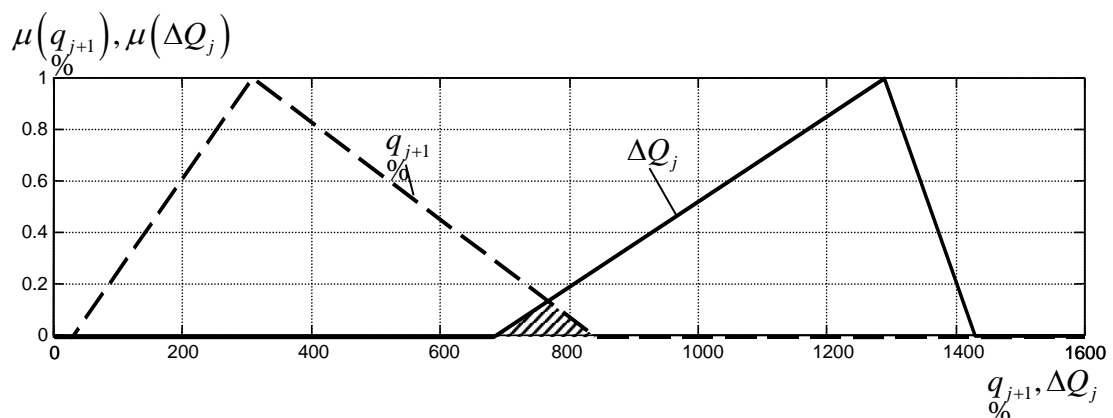
В даному розділі проведено детальний аналіз літературних джерел який показує, що на сучасному етапі розвитку методів математичного моделювання, точних і евристичних алгоритмів оптимізації та прогресивних інформаційних технологій не існує універсального підходу, що дозволяє формувати моделі і алгоритми для формалізації та розв’язку задач планування логістичних

маршрутів і траєкторій в умовах невизначеності. Більшість з них використовуються в цей час тільки для вузького класу задач. У зв'язку з великим обсягом ітераційних обчислювальних процесів та альтернативних варіантів при розв'язанні задач як попереднього планування, так і планування маршрутів та траєкторій в масштабі реального часу, методи формування та реалізації алгоритмів їх оптимізації в умовах невизначеності вимагають суттєвих обчислювальних і часових ресурсів.

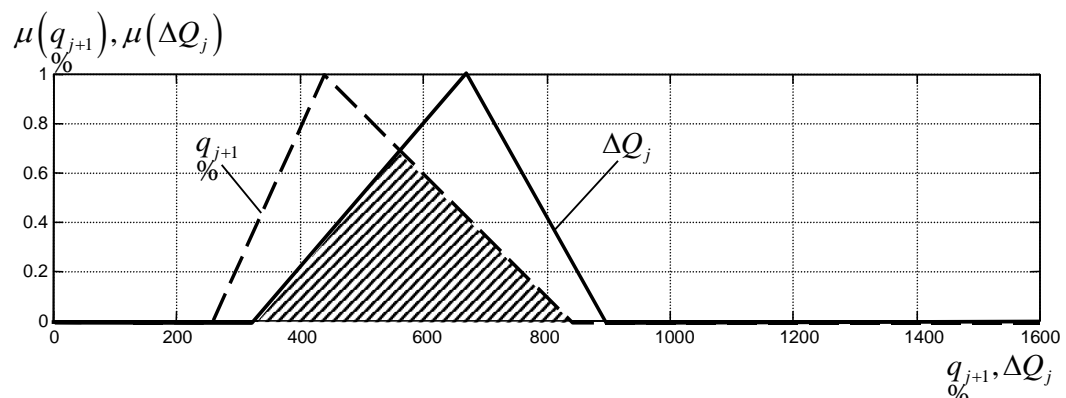
РОЗДІЛ 2. ФОРМАЛЬНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПОБУДОВИ ЛОГІСТИЧНИХ МАРШРУТІВ І ТРАЄКТОРІЙ

2.1 Фаззифікація сигналів в експертних системах підтримки прийняття рішень

Однією з найважливіших проблем при проектуванні нечітких експертних СППР є синтез алгоритмів фаззифікації якісних сигналів, тобто сигналів, що задаються нечіткими множинами для відповідних лінгвістичних термів. При цьому процес фаззифікації базується на визначенні ступеня відповідності нечіткого сигналу відповідному нечіткому лінгвістичному терму. Ступінь відповідності характеризує максимальне значення функції належності перетину (пунктирна лінія на рис. 2.1) нечітких множин відповідного лінгвістичного терма та вхідного сигналу:



a)



б)

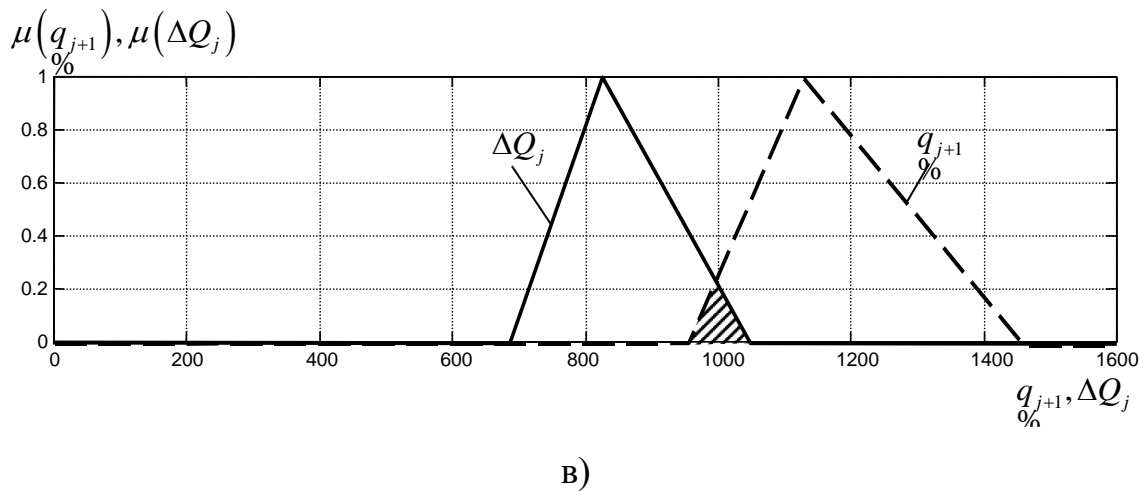


Рис. 2.1. Варіанти конфліктних ситуацій

В подальшому більш детально розглянемо процес фаззифікації якісних координат, що спирається на використання трикутної форми для представлення функцій належності нечітких вхідних сигналів $\mu_{A_{0\%}}(x)$ та лінгвістичних термів $\mu_{B_i}(x)$. Трикутна форма функцій належності в значній мірі відповідає представленню значень якісних вхідних сигналів реальних процесів. Зокрема, при розробці нечітких СППР для оптимізації процесів планування суднових бункерувальних операцій (БО) необхідно враховувати, що остаточні значення замовлень уточнюються тільки при прибутті танкера в конкретну точку морської акваторії для проведення БО на конкретному судні. Отже СППР повинна розв'язувати задачу планування маршрутів танкерів з урахуванням вантажомісткості танкерів D та нечітких замовлень, що на основі попереднього досвіду та іншої апріорної інформації може бути представлена експертами у вигляді трикутних чисел типу $A_{0\%} = (a_1, a_2, a_3)$, де a_1, a_2, a_3 - параметри трикутної ФН.

На рис. 2.2 показано процес фаззифікації якісної інформації для лінгвістичних термів трикутної форми, де позначено: $A_{0\%} = (a_1, a_2, a_3)$ – вхідний сигнал якісного типу, функція належності якого визначається залежністю

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & \text{якщо } x \leq a_1 \\ \frac{x-a_1}{a_2-a_1}, & \text{якщо } a_1 < x < a_2 \\ \frac{a_3-x}{a_3-a_2}, & \text{якщо } a_2 < x < a_3 \\ 0, & \text{якщо } x \geq a_3 \end{cases};$$

$B_{\alpha} = (b_1, b_2, b_3)$ - лінгвістичний терм трикутної форми з функцією належності

$$\mu_{\tilde{B}}(x) = \begin{cases} 0, & \text{якщо } x \leq b_1 \\ \frac{x-b_1}{b_2-b_1}, & \text{якщо } b_1 < x < b_2 \\ \frac{b_3-x}{b_3-b_2}, & \text{якщо } b_2 < x < b_3 \\ 0, & \text{якщо } x \geq b_3 \end{cases}.$$

Слід відмітити, що α -перерізи нечітких множин \tilde{A} та \tilde{B} позначаються як

$$A_{\alpha} = [a_1(\alpha), a_3(\alpha)], \quad B_{\alpha} = [b_1(\alpha), b_3(\alpha)].$$

Визначимо значення α_1 та α_2 в точках перетину нечітких множин \tilde{A} та \tilde{B} :

а) для α_1 :

$$\alpha_1 = \frac{x-a_1}{a_2-a_1}, \quad x = (a_2-a_1)\alpha_1 + a_1, \quad \text{з іншого боку } \alpha_1 = \frac{b_3-x}{b_3-b_2},$$

$$x = b_3 - (b_3-b_2)\alpha_1, \quad \text{отже } (a_2-a_1)\alpha_1 + a_1 = b_3 - (b_3-b_2)\alpha_1, \quad \alpha_1 = \frac{b_3-a_1}{(a_2-a_1) + (b_3-b_2)}$$

б) для α_2 :

$$\alpha_2 = \frac{a_3-x}{a_3-a_2}, \quad x = a_3 - (a_3-a_2)\alpha_2, \quad \text{з іншого боку}$$

$$\alpha_2 = \frac{b_3 - x}{b_3 - b_2}, \quad x = b_3 - (b_3 - b_2)\alpha_2, \text{ отже}$$

$$a_3 - (a_3 - a_2)\alpha_2 = b_3 - (b_3 - b_2)\alpha_2, \quad \alpha_2 = \frac{a_3 - b_3}{(a_3 - a_2) - (b_3 - b_2)}$$

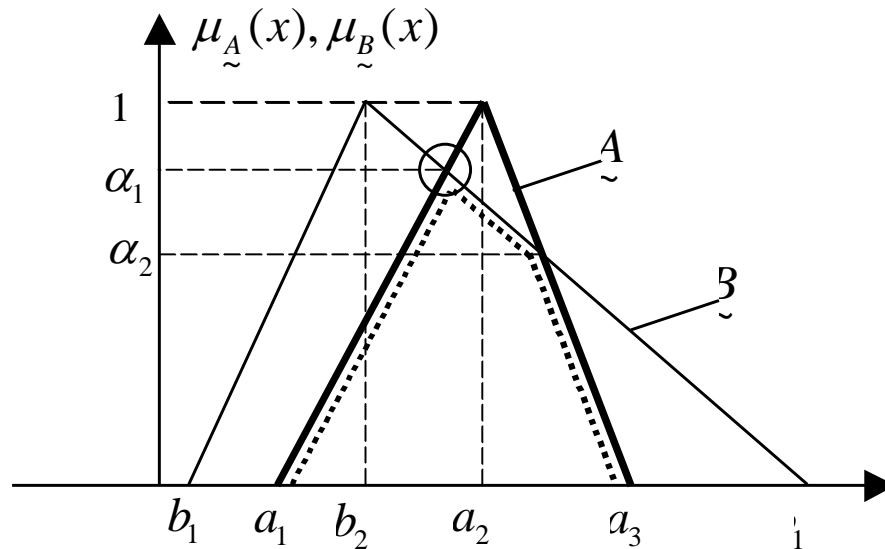


Рис. 2.2. Фазифікація якісних вхідних сигналів з ФН трикутної форми

Враховуючи різний якісний характер вхідних сигналів сформуємо алгоритми для визначення α -перерізів та результуючих функцій належності $\mu_{\tilde{A}(\vee)\tilde{B}}(x)$, що визначають рівень належності якісного вхідного сигналу \tilde{A} ($A_\alpha = [a_1(\alpha), a_3(\alpha)]$) до відповідного лінгвістичного терму \tilde{B} ($B_\alpha = [b_1(\alpha), b_3(\alpha)]$).

Сформуємо узагальнену блок-схему (рис. 2.3) алгоритму фазифікації якісних вхідних сигналів з функціями належності трикутної форми.

З метою підтвердження ефективності застосування в нечітких СППР алгоритмів фазифікації (2.1) здійснено моделювання процесів фазифікації нечітких замовлень в СППР для оптимізації суднових БО в територіально розподілених портах на основі імітаційних моделей. При цьому: $D = 1000$ т, а при формуванні параметрів нечітких замовлень використовувались процедури

формування послідовностей випадкових чисел v_j з рівномірним законом розподілу $v_j \in [0, 1], (j = 1, 2, \dots, 45)$.

$$\mu_B\left(\frac{A}{\square}\right) = \begin{cases} 0, & \text{при умові } (b_3 < a_1) \\ \frac{(a_3 - b_1)}{(b_2 - b_1) + (a_3 - a_2)}, & \text{при умові } [(b_2 > a_2) \text{ AND } (a_3 \geq b_1)] \\ 1, & \text{при умові } (a_2 = b_2) \\ \frac{b_3 - a_1}{(a_2 - a_1) + (b_3 - b_2)}, & \text{при умові } [(b_2 < a_2) \text{ AND } (b_3 \geq a_1)] \\ 0, & \text{при умові } (a_3 < b_1) \end{cases} \quad (2.1)$$

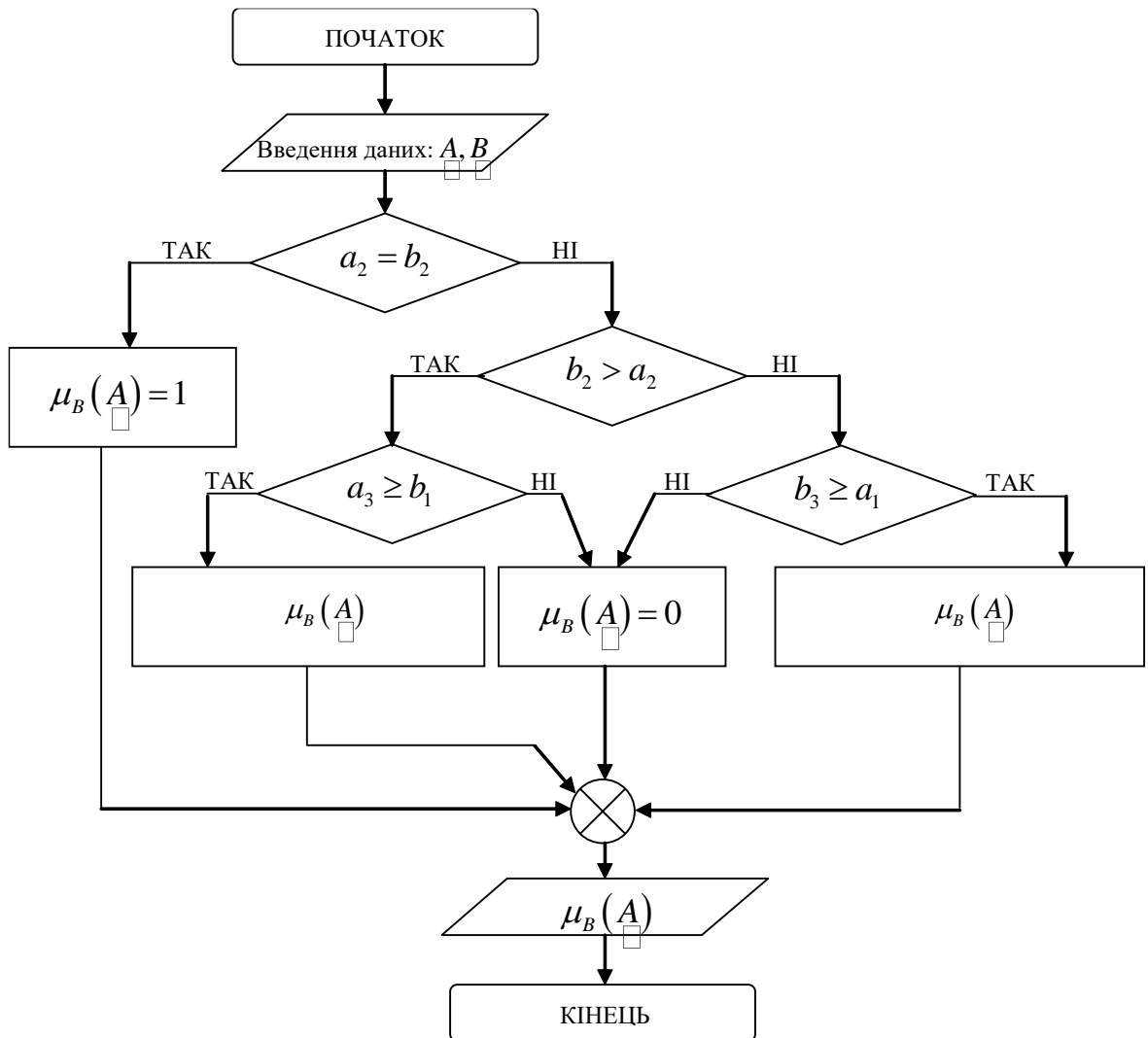


Рис. 2.3. Узагальнений алгоритм фазифікації

Процеси фаззифікації нечітких замовлень A_{oz}^k здійснено для п'яти лінгвістичних термів (рис. 2.4) B^1 : "Низький", B^2 : "Нижче середнього", B^3 : "Середній", B^4 : "Вище середнього", B^5 : "Високий".

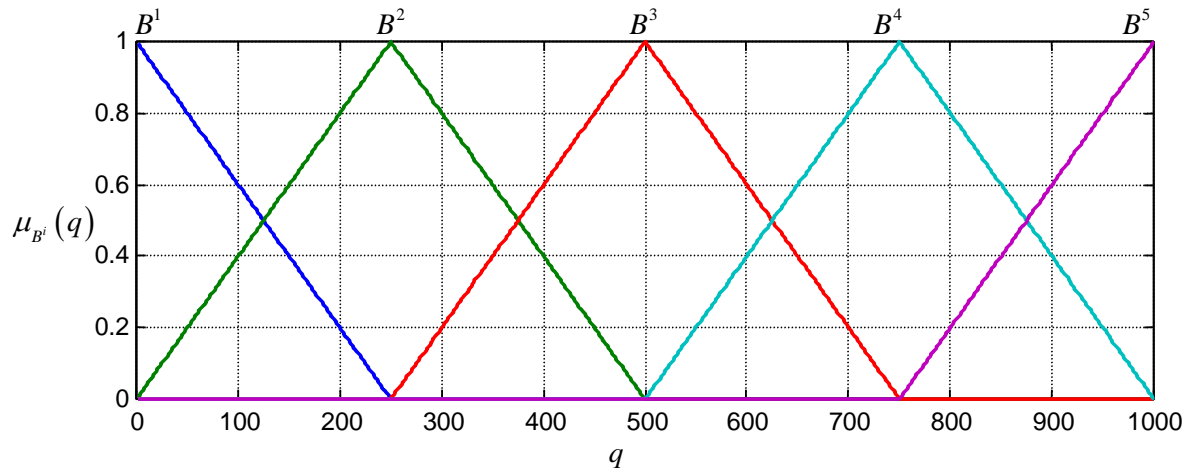


Рис. 2.4. Шкала лінгвістичних термів

Даний алгоритм фаззифікації якісної інформації та блок-схема його реалізації можуть бути універсальною складовою будь-якої нечіткої СППР, база правил якої ґрунтується на лінгвістичних термах з функціями належності трикутної форми. Алгоритм фаззифікації враховує всі можливі комбінації параметрів нечітких трикутних чисел та лінгвістичних термів з трикутними функціями належності. Застосування запропонованих алгоритмів фаззифікації в нечітких СППР для оптимізації БО, як буде показано нижче, є ефективним для обох режимів – попереднього планування маршрутів танкерів (off-line) та планування маршрутів танкерів в реальному часі (on-line).

Задача фаззифікації відносних якісних вхідних координат, що формуються як результат ділення двох трикутних чисел розкладається на дві підзадачі: визначення ФН нечіткої множини – результату ділення та розрахунок точок перетину згаданої ФН з лінгвістичними термами відповідної вхідної координати НЕС. Знаходження аналітичних виразів для фаззифікації якісних сигналів такого типу дозволить значно підвищити обчислювальну ефективність алгоритмів, що є особливо важливим для задачі планування маршрутів, так як

звертання до НЕС в цьому випадку знаходиться в тілі глибокого циклу і виконується багато разів при оцінці вузлів маршруту.

2.2 Структура нечіткої експертної системи для прийняття рішень на основі нечітких даних

Розробка програмного забезпечення системи підтримки прийняття рішень (СППР) для управління складними процесами і об'єктами в умовах невизначеності, що базується на застосуванні теорії нечітких множин та нечіткої логіки, як правило, супроводжується:

а) вибором вхідних координат, що найбільш суттєво впливають на процес прийняття рішень, наприклад шляхом побудови їх ранжованого ряду і вибором обмеженої кількості найбільш суттєвих факторів;

б) вибором кількості лінгвістичних термів та форм їх ФН, формуванням алгоритмів фазифікації вхідних координат, що можуть характеризуватися як кількісними, так і якісними величинами;

в) синтезом нечіткої бази правил;

г) вибором та розробкою алгоритмів агрегації, активації та акумуляції для формування вихідного сигналу системи підтримки прийняття рішень у вигляді нечіткої множини з відповідною ФН;

д) розробкою алгоритму формування рішення шляхом застосування алгоритмів дефазифікації або порівняння вихідних нечітких сигналів системи підтримки прийняття рішень з еталонними лінгвістичними термами, кожний з яких відповідає одному з варіантів альтернативних рішень.

Блок-схему процесу формування маршрутів і траєкторій в умовах невизначеності на основі нечіткої експертної системи (НЕС) наведено на рис. 2.5.

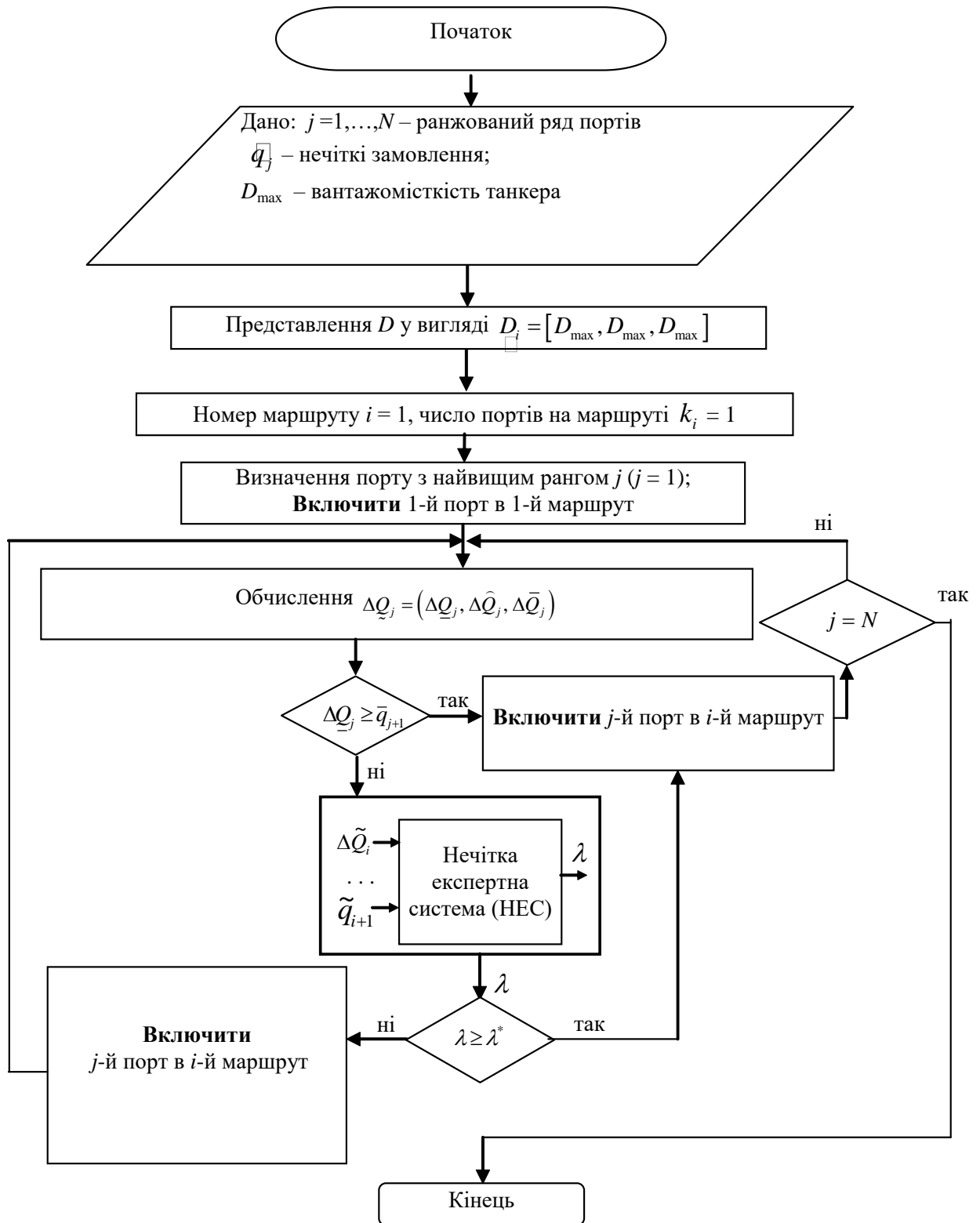


Рис. 2.5. Алгоритм формування маршрутів в умовах невизначеності з застосуванням НЕС

При плануванні маршрутів шляхом послідовного включення відповідних портів до відповідного маршруту для кожного порту здійснюється перевірка

умови перетину нечіткої множини $q_{j+1} = (\underline{q}_{j+1}, \overset{\vee}{q}_{j+1}, \bar{q}_{j+1})$, що характеризує замовлення $(j+1)$ -го порту-претендента, з нечіткою множиною $\Delta Q_{\%j} = (\Delta \underline{Q}_j, \Delta \overset{\vee}{Q}_j, \Delta \bar{Q}_j)$, що характеризує кількість залишкового бункерувального палива на танкері в j -му порту. Виконання умови $\Delta \underline{Q}_j \geq \bar{q}_{j+1}$ гарантує ефективність проведення бункерування суден-замовників у $(j+1)$ -му порту. При цьому нечітка модель $\Delta Q_{\%j} = (\Delta \underline{Q}_j, \Delta \overset{\vee}{Q}_j, \Delta \bar{Q}_j)$ залишків бункерувального палива після проведення операцій у j -му порту формується за наступним алгоритмом (наприклад, при плануванні першого маршруту):

$$\Delta Q_{\%j} = \left(D - \sum_{s=1}^j \bar{q}_s, D - \sum_{s=1}^j \overset{\vee}{q}_s, D - \sum_{s=1}^j \underline{q}_s \right)$$

Якщо вищезгадана умова не виконується, то це означає, що в процесі прийняття рішень виникає конфліктна ситуація, тобто нечітка множина $q_{j+1} = (\underline{q}_{j+1}, \overset{\vee}{q}_{j+1}, \bar{q}_{j+1})$, яка характеризує нечітке замовлення $(j+1)$ -го порту, перетинається з нечіткою множиною $\Delta Q_{\%j} = (\Delta \underline{Q}_j, \Delta \overset{\vee}{Q}_j, \Delta \bar{Q}_j)$, ступінь перетину яких можна характеризувати таким параметром, як рівень сатисфакції λ_{j+1} . Параметр λ_{j+1} формується на виході НЕС, наприклад, у вигляді нелінійної залежності

$$\lambda_{j+1} = F_{\text{НБП}}(q_{j+1}, \Delta Q_{\%j}),$$

де $q_{j+1}, \Delta Q_{\%j}$ - якісні вхідні координати нечіткої експертної системи; $F_{\text{НБП}}(q_{j+1}, \Delta Q_{\%j})$ - функція, що реалізує нечіткий логічний висновок на основі бази правил (НБП), сформованої з використанням нечіткої логіки, та алгоритмів фаззифікації, агрегації, активації, аккумуляції і дефаззифікації.

Узагальнену структуру нечіткої експертної системи для прийняття рішень на основі оцінки рівня можливої сатисфакції нечітких замовлень наведено на рис. 2.6, а алгоритм нечіткого логічного виводу представлено на рис. 2.7.

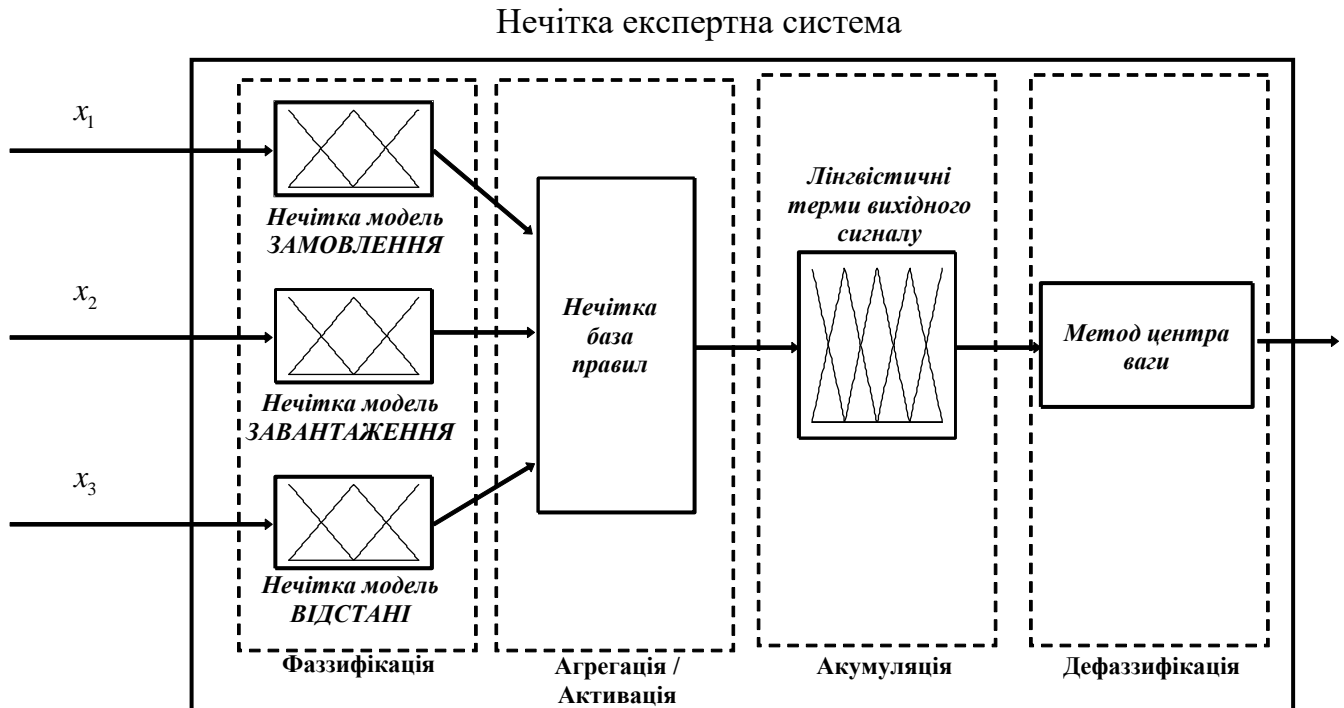


Рис. 2.6. Структура нечіткої експертної системи

При виконанні умови $\lambda_{j+1} \geq \lambda^*$ $(j+1)$ -й порт включається до відповідного маршруту, а при її порушенні – планується повернення танкера в базовий порт (*BASE*), де $\lambda^* \in [0, 1]$ – критичний рівень сатисфакції при прогнозуванні можливості задоволення замовлення в $(j+1)$ -му порту. Якщо танкер повертається в *BASE*, то формування поточного маршруту завершується і починається формування нового маршруту.

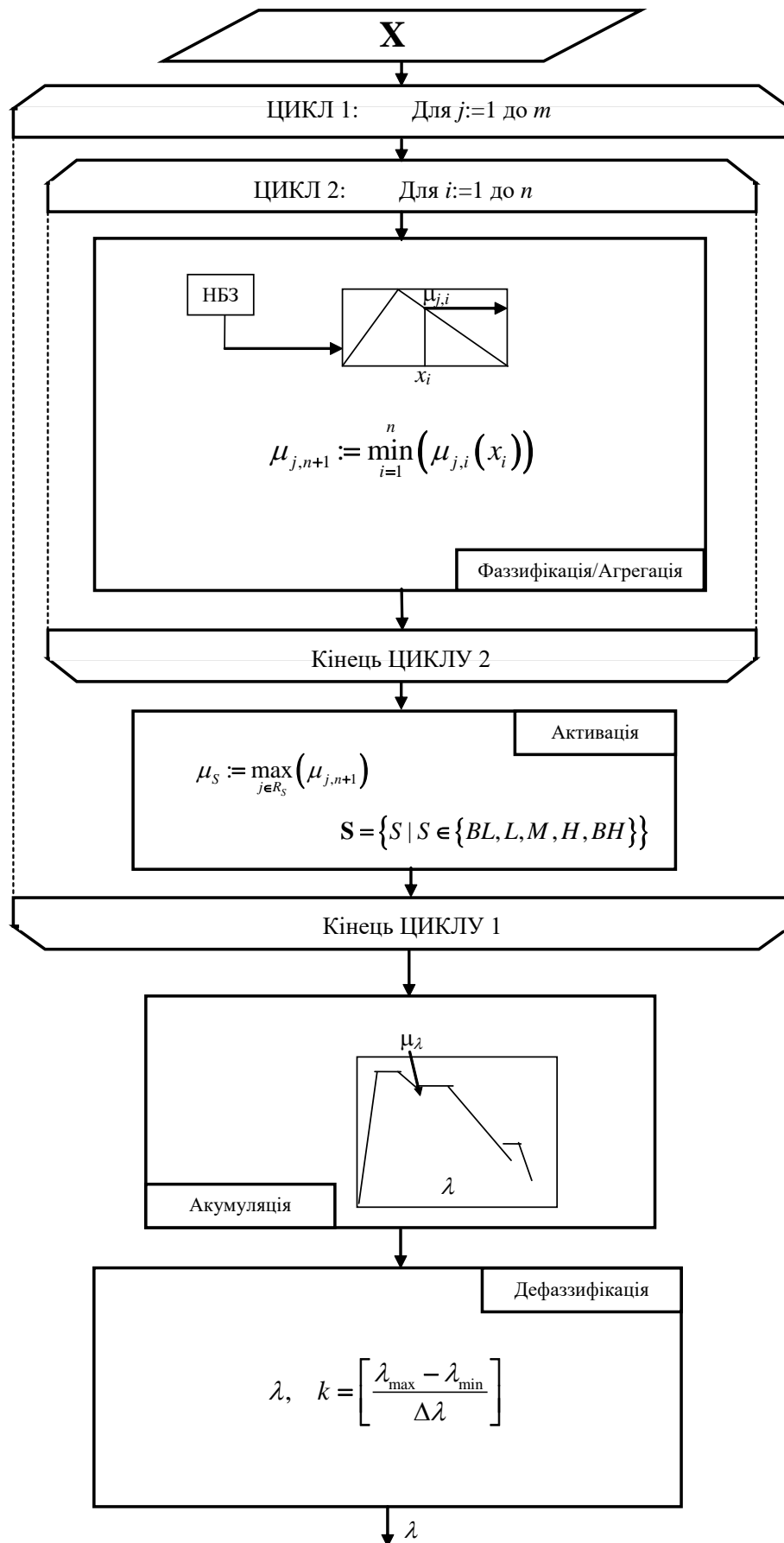


Рис. 2.7. Алгоритм нечіткого логічного виводу на основі НБЗ

Вибір значення критичного рівня сатисфакції λ^* являє собою задачу для самостійного дослідження. При виборі $\lambda^* = 0$ в результаті планування маршрутів формується тільки один маршрут, до якого входять всі порти-замовники. При виборі $\lambda^* = 1$ число маршрутів відповідає числу портів і до складу кожного маршруту входить тільки один порт. Оптимальне значення параметру λ^* , визначається в межах інтервалу $[0,4; 0,65]$.

2.3 Алгоритми агрегації нечітких множин та нечітка база правил на основі лінгвістичних термів

Розглянемо алгоритми агрегації нечітких множин, що застосовуються для STP, VRP CVRP в умовах невизначеності, в т.ч. з використанням НЕС.

ФН $\mu_{\mathcal{Z}}(C, x)$ визначає ступінь сатисфакції ЛПР відносно величини цільової функції $\mathcal{Z}(C, x)$ для конкретної STP. Ступінь сатисфакції i -го обмеження стосовно вантажомісткості є величиною функції належності $\mu_{\mathcal{Q}_i^m}(x)$:

$$\mu_{\mathcal{Q}_i^m}(x) = \begin{cases} \frac{\sum_{j=1}^n x_{ij} - \underline{Q}_i^m}{\overline{Q}_i^m - \underline{Q}_i^m}, & \underline{Q}_i^m \leq \sum_{j=1}^n x_{ij} < \overline{Q}_i^m \\ 0, & \sum_{j=1}^n x_{ij} < \underline{Q}_i^m \cup \sum_{j=1}^n x_{ij} > \overline{Q}_i^m \\ \frac{\overline{Q}_i^m - \sum_{j=1}^n x_{ij}}{\overline{Q}_i^m - \underline{Q}_i^m}, & \underline{Q}_i^m \leq \sum_{j=1}^n x_{ij} \leq \overline{Q}_i^m \end{cases}, \quad i=1,2,\dots,m.$$

Значення ФН $\mu_{\mathcal{Q}_i^m}(x)$ є оціночною функцією, яка показує, що загальний попит портів-споживачів (наприклад, для порта-постачальника 2) не перевищує вантажну спроможність \underline{Q}_2^m всіх суден, що обслуговують відповідний порт.

Ступінь сатисфакції обмежень попиту є значенням функції належності $\mu_{\underline{Q}_j^n}(x)$, яке відповідає рівню співзгодження між вантажним попитом \underline{Q}_j^n j -го порта ($j=1,2,\dots,n$) та загальною кількістю вантажу $\sum_{i=1}^m x_{ij}$ ($i=1,2,\dots,m$), який повинен бути переміщений в j -й порт з усіх портів-постачальників

$$\mu_{\underline{Q}_j^n}(x) = \begin{cases} \frac{\sum_{i=1}^m x_{ij} - \underline{Q}_j^n}{\underline{Q}_j^n - \underline{Q}_j^n}, & \underline{Q}_j^n \leq \sum_{i=1}^m x_{ij} < \bar{Q}_j^n \\ 0, & \sum_{i=1}^m x_{ij} < \underline{Q}_j^n \cup \sum_{i=1}^m x_{ij} > \bar{Q}_j^n, \\ \frac{\bar{Q}_j^n - \sum_{i=1}^m x_{ij}}{\bar{Q}_j^n - \underline{Q}_j^n}, & \bar{Q}_j^n \leq \sum_{i=1}^m x_{ij} \leq \bar{Q}_j^n \end{cases} \quad (j=1,2,\dots,n).$$

Згідно практики ЛПР зацікавлена в знаходженні “компромісного рішення” для конкретної STP та повинна мати алгоритм для пошуку таких “компромісних рішень”. Ступінь сатисфакції ЛПР стосовно розв’язку всієї STP – це $\lambda(x)$. Цей параметр $\lambda(x)$ для вищезазначеного типу “компромісних рішень” можна формувати наступним чином

$$\lambda(x) = \min_{x \in R^+ \cup \{0\}} \left(\mu_{Z_0}(C, x), \mu_{Q^{mn}}(x) \right),$$

$$\text{де } \mu_{Q^{mn}}(x) = \min_{x \in R^+ \cup \{0\}} \left(\mu_{\underline{Q}_1^n}(x), \mu_{\underline{Q}_2^n}(x) \right); \mu_{\underline{Q}_j^n}(x) = \min_{x \in R^+ \cup \{0\}} \left(\mu_{\underline{Q}_1^n}(x), \mu_{\underline{Q}_2^n}(x), \dots, \mu_{\underline{Q}_m^n}(x) \right);$$

$$\mu_{\underline{Q}_j^n}(x) = \min_{x \in R^+ \cup \{0\}} \left(\mu_{\underline{Q}_1^n}(x), \mu_{\underline{Q}_2^n}(x), \dots, \mu_{\underline{Q}_n^n}(x) \right).$$

Процедура агрегації, що базується на операторах усереднення, реалізує ідею вибору рішення в межах конфлікту між значенням цільової функції та обмеженнями вантажомісткості, а результуюче рішення лежить між найбільш

песимістичною верхньою границею та найбільш оптимістичною нижньою границею, або між максимальним та мінімальним значеннями функції належності агрегованих множин. Для агрегації нечітких множин використовується оператор “Нечітке Г”. Цей оператор базується на комбінації *min* та *max*-операторів з їх середньо-арифметичним значенням, що приводить до ряду успішних результатів.

Усереднюючий оператор “Нечітке Г” є дуже багатообіцяючим для STP, беручи до уваги, що кожен *j*-й контракт (субконтракт) стосовно перевезення вантажу між *j*-м портом-споживачем і *i*-м портом-постачальником дозволяє відхилення кількості перевезеного і розвантаженого вантажу в межах величини

ΔQ_{ij} , але загальне відхилення $\sum_{i=1}^m \Delta Q_{ij}^m$ має бути меншим, ніж $Q_{j_{perm}}^n$, де $Q_{j_{perm}}^n$ –

загальне припустиме відхилення для *j*-го порту.

Алгоритм агрегації “Нечітке Г” можна записати у вигляді

$$\mu_{\text{агг}} = \gamma \min_{i=(1\dots s)} \mu_i + (1-\gamma) \frac{1}{s} \sum_{i=1}^s \mu_i, \quad \gamma \in [0,1]$$

Якщо людина, яка приймає рішення (ЛПР) бажає змоделювати компенсаційний ефект, то необхідно обрати $\gamma < 1$ і наступна модель може бути визначена як:

$$\max \alpha + (1-\gamma) \frac{1}{s} \sum_{i=1}^s \alpha_i,$$

в залежності від обмежень $\alpha + \alpha_i < \mu_i(x), \quad i = 1, 2, \dots, s;$

$$\gamma \in [0,1]; \quad \alpha + \alpha_i < 1; \quad x \in R^+ \cup \{0\}, \quad \alpha, \alpha_i > 0.$$

Ієрархічна структура агрегації STP на основі оператора “Нечітке Г”

$$\lambda(x) = \lambda_{\text{агг}}(x) = \mu_{\text{агг}}(\mu_{\%}(C, x), \mu_{\text{огм}}(x)) \quad (1.21)$$

представлена на рис. 2.8.

Існують різноманітні пропозиції для визначення $\gamma_i (i=1,2,3,4)$, але альтернативою є можливість ЛПР корегувати параметри $\gamma_i (i=1,2,3,4)$, використовуючи апостеріорні дані та процедури оптимізації і навчання, що можуть дати найкраще рішення x^* для найкращих параметрів $\gamma_i^* (i=1,2,3,4)$.

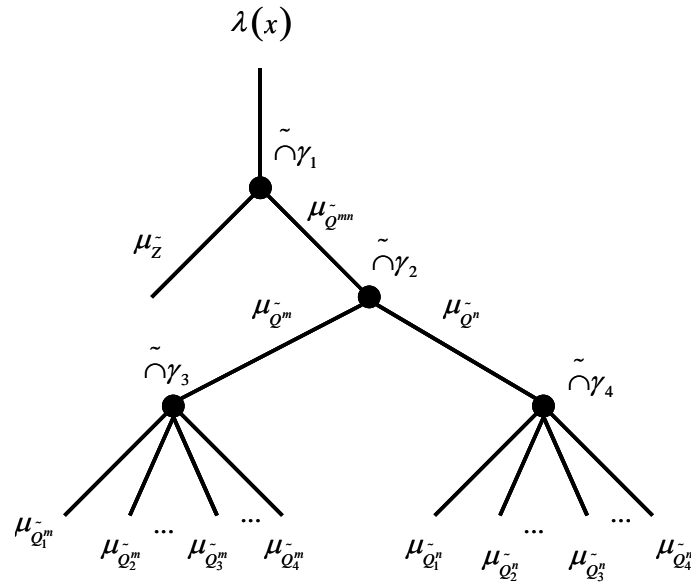


Рис. 2.8. Ієрархічна структура алгоритму агрегації

Якщо ЛПР не моделює компенсаційні ефекти, то обирається $\gamma_i = 1 (i=1,2,3,4)$ і максимізованим рішенням для зазначеної проблеми (рис.2.5) є x^* , що забезпечує максимальне значення функції належності

$$\mu(C, x) = \min_{x \in R^+ \cup \{0\}} \left(\mu_{\gamma_0}(C, x), \mu_{Q^{gm}}(x) \right).$$

Фрагмент НБЗ наведено на рис. 2.7. Структура та кількість правил в НБЗ залежать від:

- кількості I_x вхідних змінних $x_i, i=1 \dots I_x$;
- кількості J_y вихідних змінних $y_j, j=1 \dots J_y$;

- кількості S_{x_i} лінгвістичних термів $T_{x_i}^s, s = 1 \dots S_{x_i}$ вхідних змінних $x_i, i = 1 \dots I_x$;
- кількості S_{y_j} лінгвістичних термів $T_{y_j}^s, s = 1 \dots S_{y_j}$ вихідних змінних $y_j, j = 1 \dots J_y$;
- кількості ієрархічних рівнів НБЗ;
- застосування алгоритмів мінімізації кількості правил в НБП на етапах попереднього навчання НЕС.

Проектування НБЗ здійснюється експертами – спеціалістами в даній предметній області. Наприклад, при $S_{x_i} = 3$ лінгвістичні терми $T_{x_i}^s, s = 1 \dots 3$ вибираються з множини $\{L, M, H\}$, тобто $T_{x_i}^s \in \{L, M, H\}, s = 1 \dots 3$, а при $S_{y_j} = 5$ лінгвістичні терми $T_{y_j}^s, s = 1 \dots 5$ вибираються з множини $\{VL, L, M, H, VH\}$ або з множини $\{L, LM, M, HM, H\}$, тобто $T_{y_j}^s \in \{VL, L, M, H, VH\}, s = 1 \dots 5$ або $T_{y_j}^s \in \{L, LM, M, HM, H\}, s = 1 \dots 5$, де лінгвістичні терми позначено, як: VL – “ДУЖЕ НИЗЬКИЙ”; L – “НИЗЬКИЙ”; LM – “НИЖЧЕ СЕРЕДНЬОГО”; M – “СЕРЕДНІЙ”; HM – “ВИЩЕ СЕРЕДНЬОГО”; H – “ВИСОКИЙ”; VH – “ДУЖЕ ВИСОКИЙ”. Кожне окреме r -те правило ($r = 1 \dots K$) в НБП має таку структуру:

$$\text{IF } (x_1 = T_{x_1}^{sr}) \text{ AND } (x_2 = T_{x_2}^{sr}) \dots \text{ AND } (x_i = T_{x_i}^{sr}) \dots \text{ AND } (x_{I_x} = T_{x_{I_x}}^{sr}) \\ \text{THEN } (y_1 = T_{y_1}^{sr}) \text{ AND } (y_2 = T_{y_2}^{sr}) \dots \text{ AND } (y_j = T_{y_j}^{sr}) \dots \text{ AND } (y_{J_y} = T_{y_{J_y}}^{sr}),$$

де оператори AND в НЕС реалізуються на основі одного з вищеобговорених алгоритмів агрегації нечітких множин.

При $J_y = 1$ НБП трансформується до наступного вигляду.

Правило 1 (RULE 1):

$$\text{IF } (x_1 = T_{x_1}^{s1}) \text{ AND } (x_2 = T_{x_2}^{s1}) \dots \text{ AND } (x_i = T_{x_i}^{s1}) \dots \text{ AND } (x_{I_x} = T_{x_{I_x}}^{s1}) \text{ THEN } (y_1 = T_{y_1}^{s1});$$

Правило 2 (RULE 2):

IF $(x_1 = T_{x_1}^{s2})$ AND $(x_2 = T_{x_2}^{s2})$... AND $(x_i = T_{x_i}^{s2})$... AND $(x_{I_x} = T_{x_{I_x}}^{s2})$ THEN $(y_1 = T_{y_1}^{s2})$;

...

Правило r (RULE r):

IF $(x_1 = T_{x_1}^{sr})$ AND $(x_2 = T_{x_2}^{sr})$... AND $(x_i = T_{x_i}^{sr})$... AND $(x_{I_x} = T_{x_{I_x}}^{sr})$ THEN $(y_1 = T_{y_1}^{sr})$;

Правило K (RULE K):

IF $(x_1 = T_{x_1}^{sK})$ AND $(x_2 = T_{x_2}^{sK})$... AND $(x_i = T_{x_i}^{sK})$... AND $(x_{I_x} = T_{x_{I_x}}^{sK})$ THEN $(y_1 = T_{y_1}^{sK})$

Для $r = 5$, наприклад, при $J_y = 1$, $I_x = 3$, $T_{x_1}^{s5} = M$, $T_{x_2}^{s5} = H$, $T_{x_3}^{s5} = L$ та $T_{y_1}^{s5} = HM$

Правило 5 (RULE 5) в НБП буде мати вигляд:

IF $(x_1 = M)$ AND $(x_2 = H)$ AND $(x_i = L)$ THEN $(y_1 = HM)$,

а відповідний алгоритм агрегації можна записати наступним чином

$$\mu_{HM_{RULE5}}(y_1) = \mu_M(x_1) \wedge \mu_H(x_2) \wedge \mu_L(x_3) = \min\{\mu_M(x_1), \mu_H(x_2), \mu_L(x_3)\},$$

де \wedge - операція визначення мінімального значення серед відповідних ФН.

2.4 Оптимізація логістичних маршрутів на основі використання імітаційних моделей

При плануванні і оптимізації логістичних маршрутів і траєкторій необхідно застосовувати ті чи інші критерії для оцінки їх ефективності. Безумовно, найбільш достовірною оцінкою буде оцінка на основі використання апостеріорної інформації, що відображає реалізацію всіх маршрутів. Разом з тим, така оцінка може використовуватись тільки для подальшого аналізу та навчання СППР з метою вибору та настроювання параметрів ФН, корекції алгоритмів планування маршрутів і траєкторій та параметрів НЕС для розв'язання наступних задач.

В таблиці 2.1 наведено приклад формування 3-х варіантів імітаційних моделей фактичних замовлень портів P_j при плануванні маршрутів танкерів для бункерування суден в умовах апріорної нечіткості замовлень портів.

Таблиця 2.1.

Нечіткі замовлення у вигляді нечітких множин

Порт	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
\underline{q}_j	41	261	6	175	278	138	48	2	307	43	90	442
\check{q}_j	145	556	100	177	478	328	94	7	710	190	156	582
\bar{q}_j	838	568	370	702	546	444	694	621	794	956	522	880
B_j^1	798	401	341	391	316	143	595	424	456	180	305	592
B_j^2	225	266	275	646	333	367	387	237	399	680	479	569
B_j^3	525	513	70	205	332	275	179	517	401	388	445	591

Оптимізацію маршрутів і траєкторій в умовах невизначеності пропонується здійснювати шляхом додаткового розв'язання задачі TSP для кожного з синтезованих в процесі розв'язку задачі CVRP маршруту.

Сформулюємо алгоритм оптимізації маршрутів танкерів в умовах невизначеності, зокрема, при плануванні бункерувальних операцій в портах, замовлення яких представлені нечіткими множинами з відповідними ФН.

Крок 1. Визначення та аналіз вхідних даних БП для одностипних танкерів (вантажомісткість танкера (D), наближені значення замовлень портів $\check{x}_j, j=1...N$, відстані між портами) та критичного параметру $\lambda_j^* = \lambda^*, j=1...N$ для НЕС.

Крок 2. Вибір форми та параметрів ФН і формування нечітких замовлень портів на основі стохастичних імітаційних моделей.

Крок 3. Формування математичної моделі для оптимізації маршрутів суден як задачі цілочисельного програмування – CVRP з урахуванням

обмежень на кількість вантажу, що може бути реалізованим на будь-якому з маршрутів.

Крок 4. Формування на основі імітаційних моделей фактичних (реальних) вхідних параметрів (замовлень), що характеризують невизначеність апіорної інформації, та розв'язок задачі математичного програмування з урахуванням наступних вимог: а) сумарна довжина маршрутів має бути мінімізована; б) танкер на маршруті заходить в кожний порт тільки один раз (крім початкового порту *BASE*, де знаходиться бункерувальна компанія); в) початком і кінцем кожного маршруту є *BASE*; г) замовлення всіх портів мають бути виконані. При використанні евристичних методів оптимізації (у випадку великої кількості портів в БП) пропонується здійснювати наступну послідовність алгоритмічних процедур:

- розв'язок задачі TSP для всієї кількості N портів без врахування вантажних обмежень (аналог реалізації БП одним танкером з необмеженою вантажомісткістю) з метою визначення послідовності обходу портів;

- розв'язок задачі CVRP для всієї кількості N портів з врахуванням вантажних обмежень танкерів та імітаційних моделей, замовлень портів;

- оптимізація сформованих маршрутів з метою додаткової мінімізації їх загальної подовженості шляхом допоміжного розв'язку задачі TSP для кожного з маршрутів, синтезованих при розв'язанні задачі CVRP на попередньому етапі. Роботу алгоритму при реалізації *Кроку 4* проілюструємо результатами моделювання процесів планування маршрутів для БП на основі початкового порту *BASE* та 18 портів-замовників $P_j, j=1...N, N=18$. В результаті розв'язання задачі TSP отримано наступний узагальнений маршрут (без вантажних обмежень):

$$BASE - P_{11} - P_{14} - P_5 - P_6 - P_{18} - P_{13} - P_4 - P_{12} - \\ - P_{17} - P_9 - P_8 - P_{10} - P_{15} - P_3 - P_{16} - P_2 - P_1 - P_7 - BASE,$$

після розв'язку задачі CVRP ($D = 1500$ т, повна вантажна програма для всіх портів – 3894 т) сформовано наступні 3 маршрути

$$R_1 = \{BASE - P_{11} - P_{14} - P_5 - P_6 - P_{18} - BASE\},$$

$$R_2 = \{BASE - P_{13} - P_4 - P_{12} - P_{17} - P_9 - P_8 - P_{10} - P_{15} - BASE\},$$

$$R_3 = \{BASE - P_3 - P_{16} - P_2 - P_1 - P_7 - BASE\},$$

довжина L_{R_i} та вантажна програма $Q_{R_i}, i=1...3$ яких відповідно складають:

$$L_{R_1}=766 \text{ миль, } L_{R_2}=677,1 \text{ миль, } L_{R_3}=222,6 \text{ миль, } L_{\Sigma}=1665,7 \text{ миль,}$$

$$Q_{R_1}=1491,7 \text{ т, } Q_{R_2}=1382 \text{ т, } Q_{R_3}=1110,3 \text{ т.}$$

Послідовність портів в оптимізованих маршрутах (в результаті додаткового розв'язку задачі TSP для кожного маршруту $R_i, i=1...3$) є зміненою:

$$R_1 = \{BASE - P_5 - P_6 - P_{14} - P_{18} - P_{11} - BASE\},$$

$$R_2 = \{BASE - P_{15} - P_{13} - P_4 - P_{12} - P_{17} - P_9 - P_8 - P_{10} - BASE\},$$

$$R_3 = \{BASE - P_1 - P_3 - P_{16} - P_2 - P_7 - BASE\},$$

а довжина їх відповідно складає

$$L_{R_1}^*=762 \text{ миль, } L_{R_2}^*=615 \text{ миль, } L_{R_3}^*=220 \text{ миль, } L_{\Sigma}^*=1597 \text{ миль.}$$

Загальна довжина оптимізованих маршрутів в порівнянні з початковими маршрутами зменшена на $L_{\Sigma} - L_{\Sigma}^* = 68,7$ миль.

Крок 5. Аналіз кількості визначених маршрутів, оцінка можливості проведення всіх бункерувальних операцій в зазначений період часу $T_{\text{БП}}$.

Крок 6. Оцінка ефективності проведення бункерувальних операцій E_{0i} на кожному i -му маршруті за критерієм $E_{0i} = \left(1 - \frac{\Delta D_i}{D}\right)$, ($i=1, 2, \dots, r$).

Крок 7. Оцінка ефективності E_0 проведення бункерувальних операцій на всіх r маршрутах БП

$$E_0 = \frac{1}{r} \sum_{i=1}^r E_{0i} = \frac{1}{r} \sum_{i=1}^r \left(1 - \frac{\Delta D_i}{D}\right)$$

Крок 8. Апостеріорний аналіз результатів планування бункерувальних операцій та відповідне коригування параметрів СППР.

2.5 Забезпечення кращої визначеності інформації в системах на основі нечіткої логіки

Зниження ступеня невизначеності вхідної інформації доцільно забезпечувати шляхом збільшення кількості пріоритетних факторів, що визначають вхідну інформацію НЕС. Вище розроблені моделі та алгоритми функціонування НЕС не завжди враховують вплив довжини маршруту на процес прийняття локальних рішень. Разом з тим ЛПП, що приймає рішення, розуміє, що немає сенсу прямувати до наступного порта-претендента, коли відстань до нього є дуже великою, а при цьому залишок вантажу у танкері є незначним.

Щоб зкоригувати результат прийняття рішення, додамо до нечіткої бази правил ще одну вхідну координату.

Для оптимізації процесу можна використовувати нечітку модель задачі маршрутизації. При рішенні такої CVRP в залежності від розмірності задачі (кількості портів) використовується один з точних або евристичних алгоритмів для розв'язання задачі комівояжера. В подальшому за допомогою НЕС з 3-ма входами та одним виходом (з різними типами ФН і числа термів) формуються маршрути, які потім оптимізуються шляхом додаткового розв'язання задачі комівояжера для кожного з запланованих маршрутів.

Беручи до уваги лінгвістичні терми L (Low – низький), M (Middle – середній), H (High – високий) та ФН трикутної форми, сформуємо варіант НБП для розв'язання задачі CVRP (де q_{next} – нечітке значення замовлення порта-претендента, $Q_{Current}$ – нечітке значення залишкового палива на танкері після обслуговування поточного порту, $delta$ – різниця в відстанях для двох варіантів маршруту – з заходом і без заходу в базовий порт, $Lambda$ – рівень сатисфакції):

1. **If** (*q-next is L*) and (*Q-Current is L*) and (*delta is L*) then (*Lambda is L*)
2. **If** (*q-next is L*) and (*Q-Current is L*) and (*delta is M*) then (*Lambda is L*)
3. **If** (*q-next is L*) and (*Q-Current is L*) and (*delta is H*) then (*Lambda is M*)
4. **If** (*q-next is L*) and (*Q-Current is M*) and (*delta is L*) then (*Lambda is M*)
5. **If** (*q-next is L*) and (*Q-Current is M*) and (*delta is M*) then (*Lambda is M*)
6. **If** (*q-next is L*) and (*Q-Current is M*) and (*delta is H*) then (*Lambda is H*)
7. **If** (*q-next is L*) and (*Q-Current is H*) and (*delta is L*) then (*Lambda is M*)
8. **If** (*q-next is L*) and (*Q-Current is H*) and (*delta is M*) then (*Lambda is H*)
9. **If** (*q-next is L*) and (*Q-Current is H*) and (*delta is H*) then (*Lambda is H*)
10. **If** (*q-next is M*) and (*Q-Current is L*) and (*delta is L*) then (*Lambda is L*)
11. **If** (*q-next is M*) and (*Q-Current is L*) and (*delta is M*) then (*Lambda is L*)
12. **If** (*q-next is M*) and (*Q-Current is L*) and (*delta is H*) then (*Lambda is M*)
13. **If** (*q-next is M*) and (*Q-Current is M*) and (*delta is L*) then (*Lambda is L*)
14. **If** (*q-next is M*) and (*Q-Current is M*) and (*delta is M*) then (*Lambda is M*)
15. **If** (*q-next is M*) and (*Q-Current is M*) and (*delta is H*) then (*Lambda is M*)
16. **If** (*q-next is M*) and (*Q-Current is H*) and (*delta is L*) then (*Lambda is M*)
17. **If** (*q-next is M*) and (*Q-Current is H*) and (*delta is M*) then (*Lambda is H*)
18. **If** (*q-next is M*) and (*Q-Current is H*) and (*delta is H*) then (*Lambda is H*)
19. **If** (*q-next is H*) and (*Q-Current is L*) and (*delta is L*) then (*Lambda is L*)
20. **If** (*q-next is H*) and (*Q-Current is L*) and (*delta is M*) then (*Lambda is L*)
21. **If** (*q-next is H*) and (*Q-Current is L*) and (*delta is H*) then (*Lambda is L*)
22. **If** (*q-next is H*) and (*Q-Current is M*) and (*delta is L*) then (*Lambda is L*)
23. **If** (*q-next is H*) and (*Q-Current is M*) and (*delta is M*) then (*Lambda is L*)
24. **If** (*q-next is H*) and (*Q-Current is M*) and (*delta is H*) then (*Lambda is M*)
25. **If** (*q-next is H*) and (*Q-Current is H*) and (*delta is L*) then (*Lambda is M*)
26. **If** (*q-next is H*) and (*Q-Current is H*) and (*delta is M*) then (*Lambda is H*)
27. **If** (*q-next is H*) and (*Q-Current is H*) and (*delta is H*) then (*Lambda is H*)

Для планування маршрутів при зниженому рівні невизначеності вхідної інформації розроблено алгоритми off-line (режим попереднього планування) і

on-line (режим оперативного планування), що забезпечують визначення поточних значень рівня сатисфакції. Розглянемо нечіткі алгоритми для режимів попереднього та оперативного планування маршрутів більш детально.

Алгоритм on-line (режим оперативного планування) реалізує ситуацію, коли замовлення портів є нечіткими числами, а залишковий вантаж є чітким. Така ситуація виникає при прийнятті рішення капітаном танкера про обслуговування порта-претендента безпосередньо в поточному порту після його обслуговування .

Алгоритм off-line (режим попереднього планування) застосовується на етапі планування маршрутів в базовому порту за умови, що замовлення портів і залишковий вантаж вважаються нечіткими числами.

Пропонується вхідні сигнали формувати у вигляді відносних величин, що також зменшує ступінь невизначеності вхідної інформації, враховуючи недоцільність заходу в порт-претендент (у випадку конфліктної ситуації), якщо він знаходиться на досить значній відстані. На рис. 2.9 схематично представлена НЕС для планування маршрутів в режимі off-line.

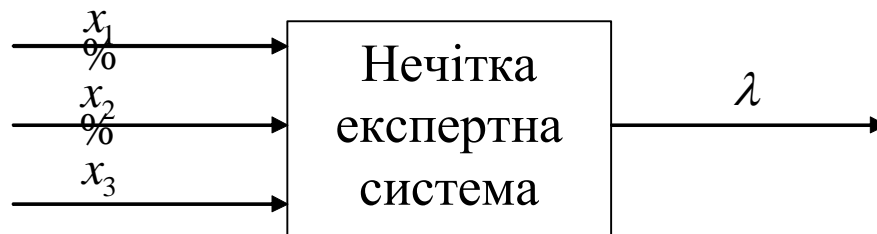


Рис. 2.9. Оцінка рівня сатисфакції в режимі off-line

Рівень сатисфакції λ для off-line режиму визначається за нечітким алгоритмом:

$$\lambda = f_{RB} \left(q_{i+1}, \Delta Q_i, D_{\max}, L_1, L_2 \right),$$

$$\lambda = f_{RB} \left(\frac{q_{i+1}}{\frac{\%}{0\%}}, \frac{\Delta Q_i}{D_{\max}}, \frac{L_1}{L_2} \right),$$

$$\lambda = f_{RB} (x_{0\%}, x_{0\%}, x_3),$$

Рівень сатисфакції λ для on-line режиму визначається за нечітким алгоритмом:

$$\lambda = f_{RB} (q_{i+1}, \Delta Q_i, D_{\max}, L_1, L_2), \quad \lambda = f_{RB} \left(\frac{q_{i+1}}{\frac{\%}{0\%}}, \frac{\Delta Q_i}{D_{\max}}, \frac{L_1}{L_2} \right),$$

$$\lambda = f_{RB} (x_{0\%}, x_2, x_3),$$

2.6 Опис алгоритмів пошуку найближчих вузлів-претендентів

В багатьох випадках, зокрема для реалізації БП, при плануванні маршрутів і траєкторій необхідно забезпечувати як мінімізацію загальної довжини маршрутів, так і максимізацію кількості розвантаженого палива на кожному з маршрутів за критеріями.

Головна ідея розв'язання цієї проблеми – розробити структуру СППР, яка пропонуватиме ЛПР декілька альтернативних рішень з метою забезпечення максимально можливого розвантаження вантажу на всіх маршрутах в цілому і на кожному з окремих маршрутів, беручи до уваги, що критерій загальної довжини всіх маршрутів повинен бути мінімізованим наскільки це можливо з економічної точки зору. Для реалізації даного підходу пропонується змінювати глибину пошуку портів-претендентів за наступними алгоритмами:

Алгоритм 1 (глибина пошуку = 0): танкер прямує до порта-претендента, якщо рівень сатисфакції $\lambda \geq \lambda^*$, інакше – танкер повертається до базового порту;

Алгоритм 2 (глибина пошуку = 1): танкер прямує до порта-претендента, якщо рівень сатисфакції $\lambda \geq \lambda^*$, інакше, перш ніж повернутися до базового порту, перевіряється наступний порт-претендент, що знаходиться найближче до поточного;

Алгоритм 3 (глибина пошуку = 2): танкер прямує до порта-претендента, якщо виконується умова $\lambda \geq \lambda^*$, інакше, перевіряються на виконання цієї умови наступні 2 порти-претенденти, що слідують (внаслідок попереднього розв'язку задачі комівояжера) за поточним портом-претендентом.

На рис. 2.10 проілюстровано запропонований метод планування маршруту танкера при виконанні БП в умовах невизначеності. Умова виконується для портів d_1, d_2, d_3 і не виконується для порту-претендента d_4 . СППР на основі НЕС з трьома входами оцінює рівень сатисфакції λ для двох (або більше) найближчих портів-претендентів d_5 та d_6 (як альтернативні версії), використовуючи інформацію про нечіткі $\Delta Q_{i, \text{off}}$ (off-line – режим) або чіткі ΔQ_i (on-line – режим) значення вантажу, який залишається на танкері, загальну вантажомісткість танкеру D_{\max} , нечіткі величини замовлень портів q_{i+1} , довжину додаткових „віртуальних” маршрутів L_1 (рис. 4.17), L_2 (рис. 4.18) та L_2' (рис. 4.19) відносно нейтральної точки – порту d_7 .

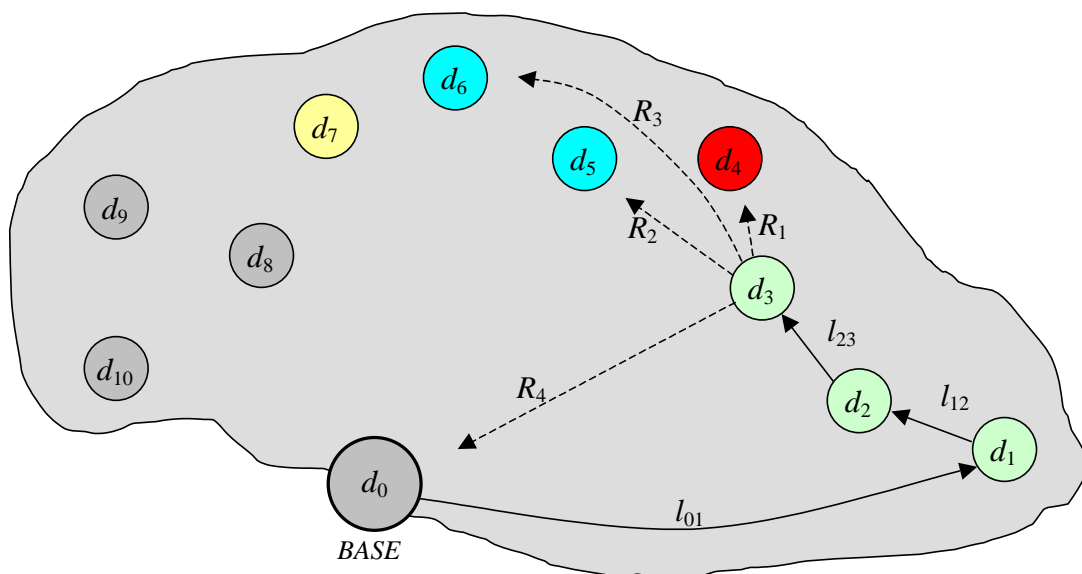


Рис. 2.10. Модель для генерації для альтернативних рішень

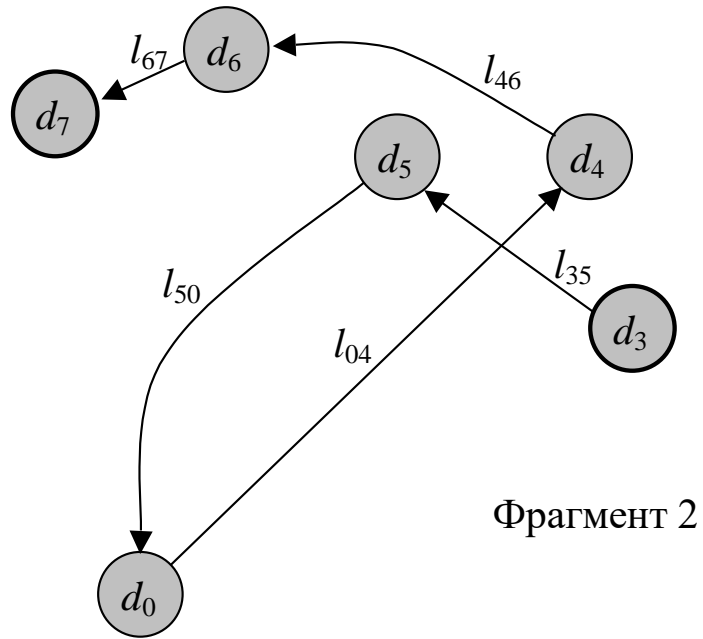


Рис. 2.11. Довжина фрагменту 2: $L_2 = l_{35} + l_{50} + l_{04} + l_{46} + l_{67}$

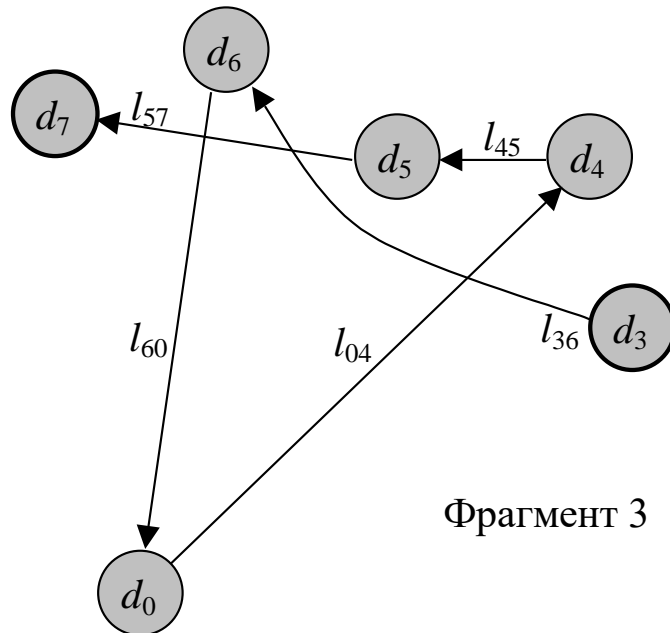


Рис. 2.12. Довжина фрагменту 3: $L_2' = l_{36} + l_{60} + l_{04} + l_{45} + l_{57}$

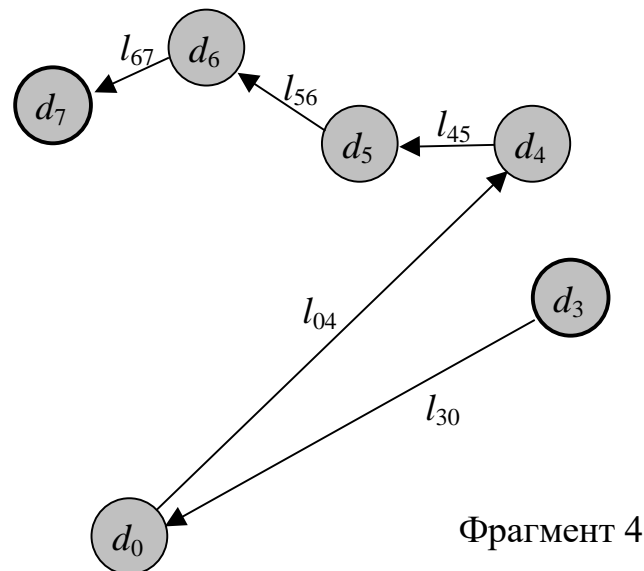


Рис. 2.13. Довжина фрагменту 4: $L_1 = l_{30} + l_{04} + l_{45} + l_{56} + l_{67}$

Таким чином, в порту d_3 ЛПР має як мінімум чотири альтернативних рішення:

- танкер прямує до порту-претендента d_4 (фрагмент 1);
- танкер прямує до наступного найближчого (після d_4) порта-претендента d_5 (фрагмент 2);
- танкер прямує до наступного найближчого (після d_5) порта-претендента d_6 (фрагмент 3);
- танкер прямує до базового порта d_0 (фрагмент 4).

Для оцінки ефективності евристичних алгоритмів з різною глибиною пошуку порта-претендента розроблено програмний продукт на основі нечіткої логіки.

2.7. Моделі, що враховують часові та інші обмеження при плануванні логістичних маршрутів

Вибір того чи іншого методу формування математичних моделей, евристичних алгоритмів та алгоритмів оптимізації маршрутів суттєво залежить від особливостей постановки задачі VRP, існуючих обмежень та припущень.

Формування маршрутів, як правило, здійснюється на основі апіорної інформації, що надається замовниками диспетчеру (оператору), який розміщується в базовому вузлі. До основних характеристик замовників слід віднести:

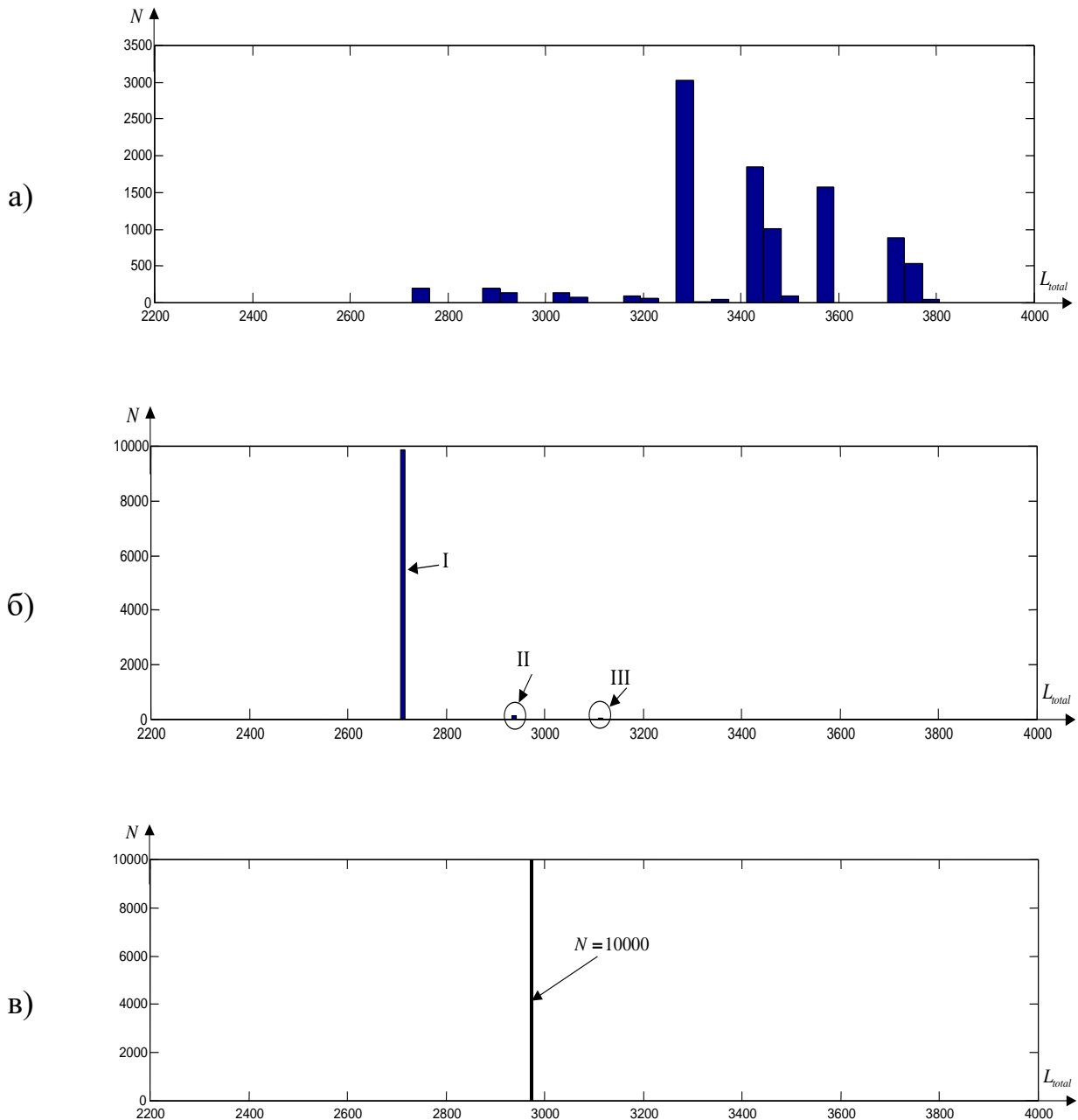


Рис. 2.14. Гістограми розподілу параметру L_{total} для стохастичних моделей

замовлень (БП № 4) при плануванні маршрутів з:

а) $\lambda^* = \lambda^*_{\min} = 0,4$; б) $\lambda^* = 0,5$; в) $\lambda^* = \lambda^*_{\max} = 0,6$

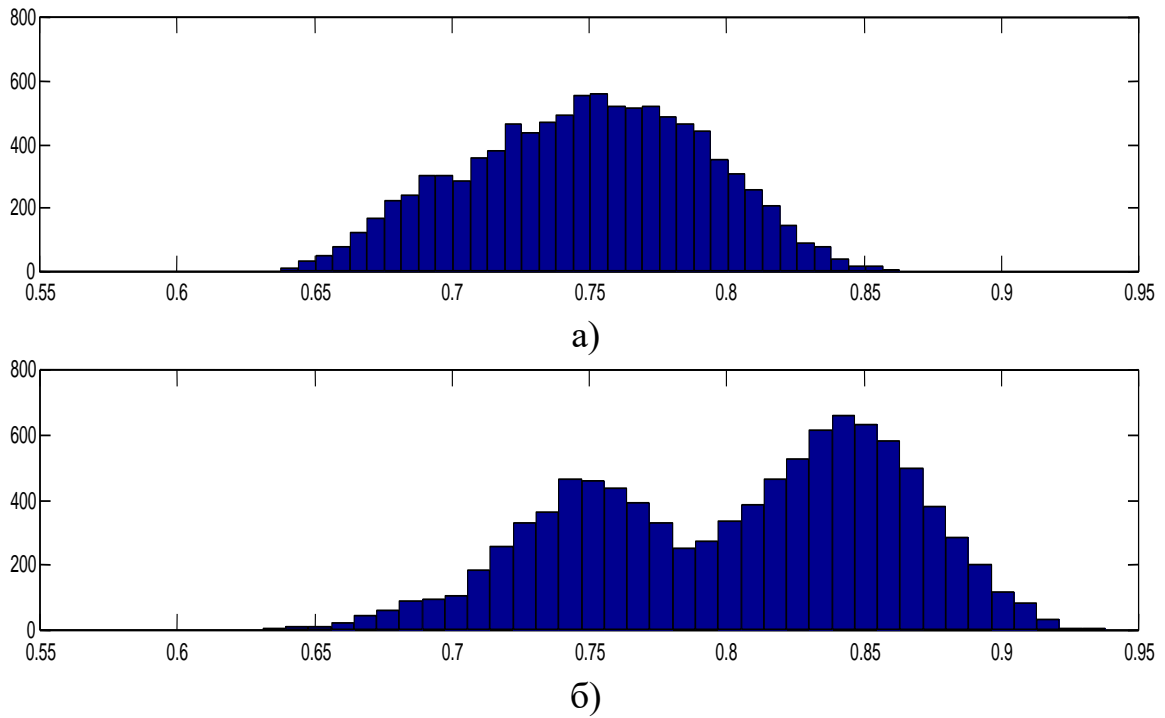


Рис. 2.15. Гістограми розподілу значень критерію E_{r-1} при плануванні маршрутів для стохастичних моделей замовлень (БП № 7, $\lambda^* = 0.5$):

а) глибина пошуку = 0 ($E_0^{av} = 0,7489$);

б) глибина пошуку = 2 ($E_0^{av} = 0,8042$)

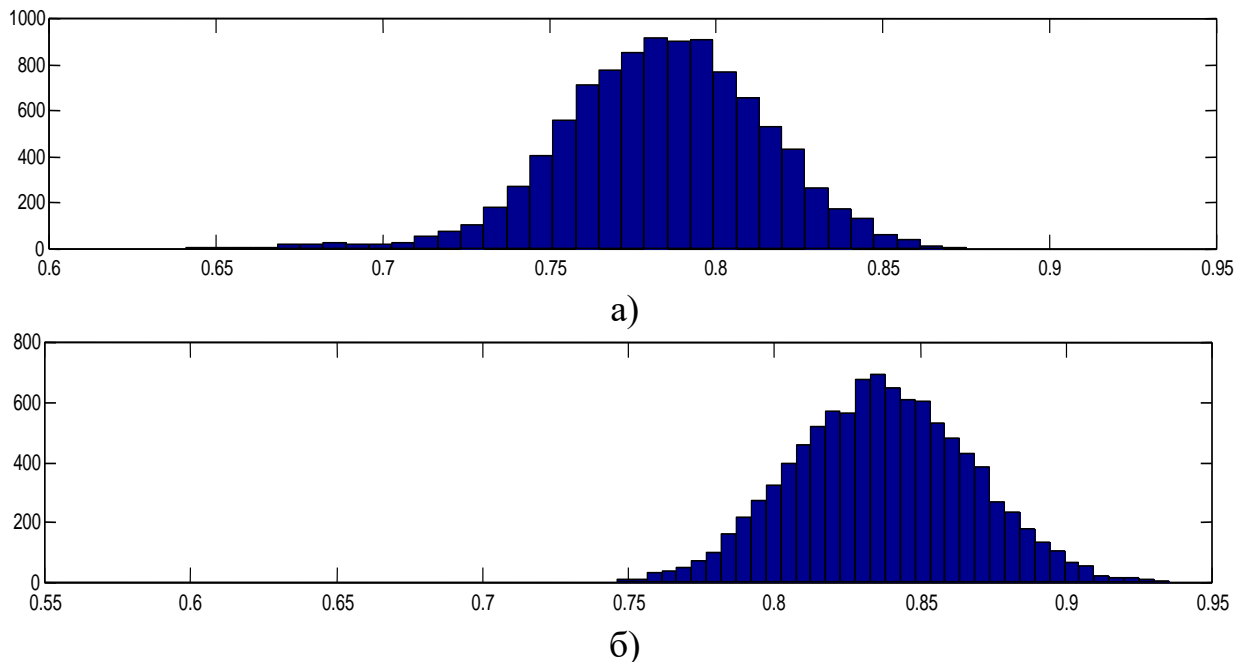


Рис. 2.16. Гістограми розподілу значень критерію E_{r-1} при плануванні маршрутів для стохастичних моделей замовлень (БП № 4, $\lambda^* = 0.5$):

а) глибина пошуку = 0 ($E_0^{av} = 0,7836$);

б) глибина пошуку = 2 ($E_0^{av} = 0,8370$)

- конкретний вузол графа, замовлення якого представляє відповідний замовник;

- кількість замовників відповідає кількості вузлів N на всіх маршрутах з урахуванням того, що кожен вузол має входити до складу тільки одного з множини всіх маршрутів;

- параметри замовлення, що має бути доставленим до відповідного вузла (в загальному випадку вантаж може характеризуватись різними типами товарів);

- часовий термін (часове вікно), на протязі якого замовлення відповідного вузла має бути виконаним;

- часовий термін, яким обмежується час проведення розвантажувальних операцій у відповідному вузлі;

- кількість транспортних одиниць, що можуть бути задіяними для реалізації всіх запланованих маршрутів при виконанні повної програми транспортних перевезень.

Розглянемо більш детально основні класи задач VRP [10], які зустрічаються в практиці планування та оптимізації маршрутів і траєкторій рухомих об'єктів.

Основною серед задач VRP є задача CVRP, тобто задача планування маршрутів для задоволення замовлень вантажу у вузлах при обмеженій вантажомісткості D транспортних одиниць, що реалізують відповідні маршрути.

Якщо замість обмежень на вантажомісткість мають місце обмеження на загальну довжину (або час реалізації) маршруту, то задачу такого класу відносять до задач DVRP (Distance-Constrained VRP). Транспортні перевезення можуть характеризуватись також двома видами обмежень: обмеженнями на вантажомісткість транспортних одиниць та загальну довжину кожного маршруту. Задачі такого класу відносять до задач DCVRP (Distance-Constrained Capacitated VRP). Якщо на маршруті передбачається два типи обслуговування замовників у вузлах, зокрема, доставка вантажу в m вузлів та завантаження транспортної одиниці на зворотному шляху вантажами інших $N - m$ замовників, то формування маршрутів здійснюється для задач класу VRPB

(VRP with Backhauls), що можуть бути як симетричними, так і асиметричними в залежності від типу матриці витрат C . Задача визначення мінімально можливої кількості транспортних одиниць для виконання всіх замовлень у відповідних вузлах складає суть задачі класу BPP (Bin Packing Problem). В деяких випадках при реалізації маршруту замовник у відповідному i -му вузлі має не тільки отримати вантаж від транспортної одиниці згідно з попереднім замовленням d_i , а й завантажити транспортну одиницю власним вантажем u_i . Методи розв'язання задач такого типу характерні для задач класу VRPPD (VRP with Pickup and Delivery). Існують також задачі формування і оптимізації маршрутів при стохастичному характері часу переміщення транспортних одиниць між вузлами (задачі типу VRPST – VRP with Stochastic Travel Times), стохастичному характері замовлень (задачі типу VRPSD – VRP with Stochastic Demands).

Особлива увага приділяється розв'язанню задачі CVRP в умовах часових обмежень на обслуговування кожного i -го вузла (в межах часових інтервалів $[a_i, b_i], i = 1 \dots N$), тобто складає суть розв'язання задачі класу VRPTW (VRP with Time Windows). Широко поширені для розв'язання такого класу задач “часові віконні” алгоритми. В деяких випадках планування маршрутів і траєкторій при реалізації БП з часовими обмеженнями необхідно здійснювати з використанням таких “часових віконних” алгоритмів.

Висновки до розділу

В даному розділі подано теоретичні положення, моделі та алгоритми для прийняття рішень в умовах невизначеності при плануванні логістичних маршрутів і траєкторій, а також наведено процес моделювання з використанням нечіткої експертної системи на основі імітаційних моделей. Обґрунтовано структуру та принципи побудови нечіткої експертної системи для прийняття рішень на основі попередньої оцінки рівня сатисфакції нечітких вхідних даних. Розглянуто клас задач VRP, що враховують часові та інші обмеження при плануванні маршрутів і траєкторій в умовах невизначеності.

РОЗДІЛ 3. ПРИКЛАДНА РЕАЛІЗАЦІЯ СИСТЕМИ ПОБУДОВИ ЛОГІСТИЧНИХ МАРШРУТІВ НА ОСНОВІ НЕЧІТКИХ МОДЕЛЕЙ

3.1 Основні концепції проекту системи підтримки прийняття рішень

Даний розділ присвячено створенню інформаційної системи для планування логістичних маршрутів та траєкторій, мета якої полягає в забезпеченні можливостей для поширення області застосування інтелектуальної системи на більш широкий клас задач логістики, що базуються на необхідності доставки ресурсів при нечітких замовленнях в вузлах.

Таким чином на етапі планування процесів розробки інтелектуальної інформаційної системи формування маршрутів і траєкторій необхідно закласти широкі можливості щодо її подальшої модернізації, сумісності при включенні як компоненти до складу більш великих проектів та адаптації до інших предметних областей. Серед існуючих парадигм програмування поставленим вимогам найбільш відповідає об'єктно-орієнтована технологія.

В результаті аналізу існуючих реалізацій об'єктно-орієнтованих мов програмування для розробки проекту обрано мову, що містить всі необхідні об'єктні концепції (поліморфізм, інкапсуляція, наслідування тощо), має ряд ефективних реалізацій та відзначається ясністю представлення алгоритмів, що також важливо для навчальних задач.

Для досягнення сформульованої мети функціонування інтелектуальної системи мають бути розв'язані задачі з розробки та реалізації:

- об'єктної моделі нечіткої експертної системи;
- об'єктної моделі розв'язання класичної задачі комівояжера за методом Clark і Wright та шляхом моделювання процесів випалювання;
- об'єктної моделі алгоритму формування оптимальних маршрутів з урахуванням невизначеності замовлень;
- об'єктної моделі представлення даних, зокрема, бази замовлень, бази замовників, бази можливих дислокацій замовників та ін.;

- об'єктної моделі для візуалізації початкових даних і результатів роботи програмного продукту, що забезпечить максимально ефективну взаємодію СППР та ЛПР і має строго відповідати наявним промисловим стандартам в цій області.

На основі сформульованих вище задач програмної системи в предметній області (планування маршрутів в умовах невизначеності) сформовано наступний перелік ключових абстракцій:

- Система планування маршрутів,
- Бункерувальна програма,
- Замовник,
- Дислокація,
- Замовлення,
- Маршрут,
- План маршрутів,
- Завантаження,
- База замовників,
- База дислокацій,
- База замовлень,
- Нечітка експертна система,
- Функція належності,
- Лінгвістична змінна,
- Лінгвістичне правило.

У відповідності до функціональних властивостей виділено наступні основні модулі інтелектуальної системи планування маршрутів:

- модуль *DataClasses*, призначений для об'єднання класів, що представляють собою елементарні об'єкти даних: Замовник, Дислокація та Замовлення;
- модуль *DataBases*, що об'єднує класи-обробники баз об'єктів даних: База замовників, База дислокацій та База замовлень;

- модуль *FuzzyClasses*, в якому згруповано всі класи, що вирішують задачі опису та функціонування нечітких експертних систем, а також здійснення арифметичних операцій над нечіткими числами: Нечітка експертна система, Функція належності, Лінгвістична змінна, Лінгвістичне правило;

- модуль *MainProgram*, який об'єднує основні класи програмної системи: Система планування маршрутів, Бункерувальна програма, Маршрут та План маршрутів. Реалізації методів даних класів містять операції по плануванню оптимальних маршрутів з урахуванням та без урахування обмежень на вантажомісткість транспортної одиниці.

3.2 Об'єктна декомпозиція задачі планування логістичних маршрутів та розробка діаграм класів

Для представлення внутрішньої структури класів та характеру семантичних взаємозв'язків між ними застосовано діаграми класів в поширеній нотації Буча (рис. 3.1), яка є найбільш інформативною та інтуїтивно зрозумілою. В програмній системі, згідно до прийнятих в Object Pascal правил формування ідентифікаторів, назви класів є іменниками або сполученнями іменників англійської мови. Всі ідентифікатори класів починаються з великої літери *T*, (наприклад, *TRule*), окремі слова в ідентифікаторі починаються з великої літери, (наприклад, *TBounkeringProgram*). Повну специфікацію імен класів, об'єктів, властивостей та операцій розроблено згідно до існуючих норм мови Object Pascal.

Розглянемо більш детально об'єктну модель вхідних даних задачі планування оптимальних маршрутів та траєкторій (рис. 3.1). Нижче подано детальний опис кожного з класів, що відносяться до моделі даних (модуль *DataClasses*).

Базовий абстрактний клас для збереження даних *TDataObject* успадковується від стандартного базового класу *TObject* і містить ключове поле *KeyId*, що служить для збереження унікального ідентифікатора об'єкта даних.

Передбачається, що вказники на об'єкти (Замовлення, Замовники та Дислокації) формуються не за фізичною адресою об'єкта в пам'яті, а за значенням ключового ідентифікатора. При необхідності фізична адреса визначається об'єктом-клієнтом шляхом запиту до відповідної бази (об'єкта-сервера). Такий механізм покликаний забезпечувати відновлюваність об'єктів "Бункерувальна програма" та "Система планування маршрутів" після перерв у роботі СППР.

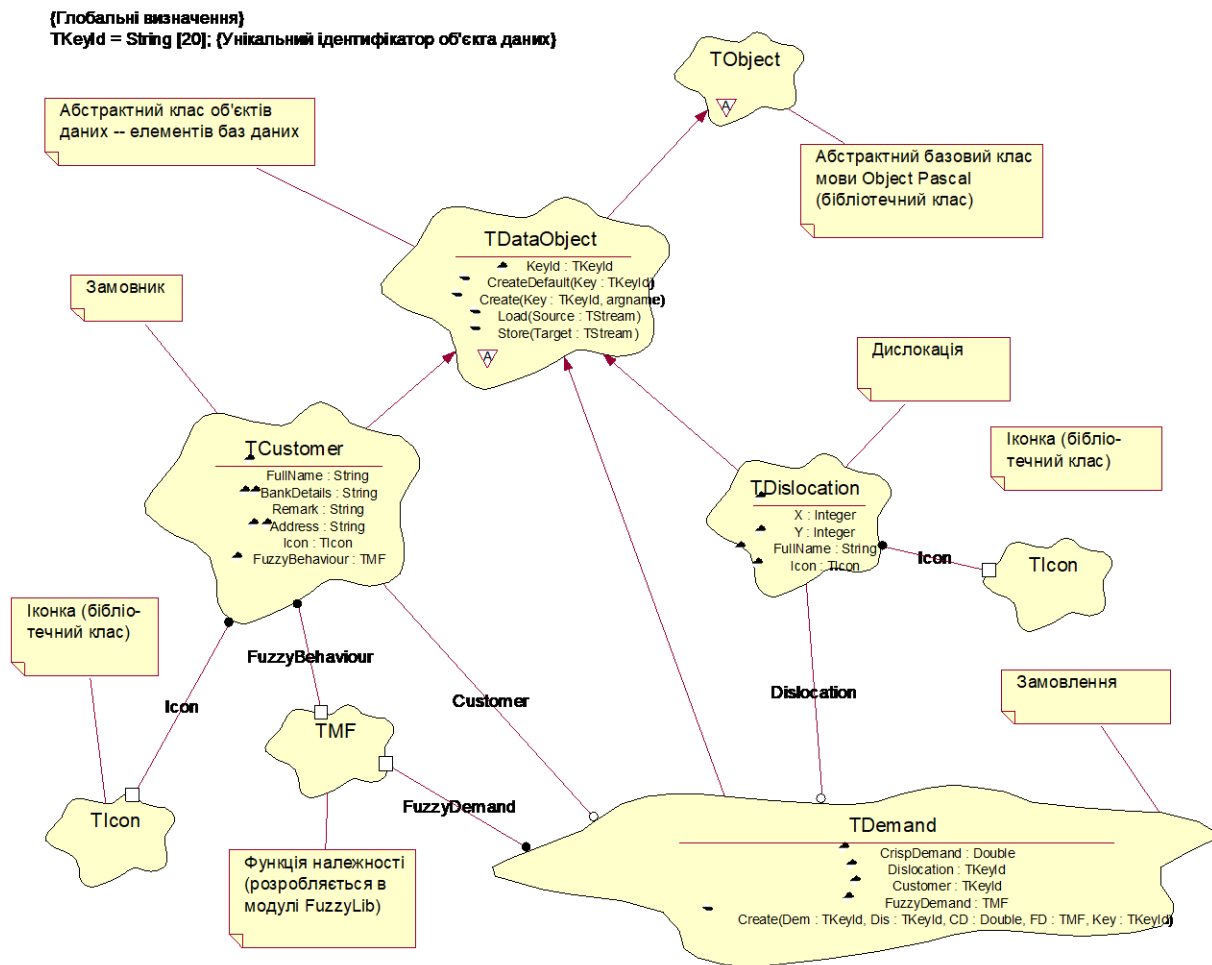


Рис. 3.1. Діаграма класів модуля *DataClasses*

Абстрактний клас *TDataObject*, включає наступні операції, що є загальними для всіх об'єктів даних:

- конструктор *Create*, що створює новий об'єкт даних із заданими параметрами;
- конструктор *CreateDefault*, що створює новий об'єкт даних з параметрами по замовчуванню;
- конструктор *Load*, що призначений для створення нового об'єкта даних, значення параметрів якого зчитуються з потоку даних;
- віртуальна процедура *Store*, що призначена для збереження стану об'єкта даних в потоці (в базовому класі процедура виводить до потоку значення поля *KeyId*).

Всі класи-нащадки повинні перевизначати поведінку перелічених операцій.

Клас "Замовник" (*TCustomer*) є нащадком класу *TDataObject* і призначається для збереження інформації про організацію (компанію), що замовляє паливо. Так як одна організація може подавати декілька замовлень в межах одної БП, то, для уникнення дублювання інформації, даний клас розділено з класом "Замовлення".

На рис. 3.1 показано структуру класу *TCustomer*, до властивостей якого відносяться: повна назва замовника, юридична адреса, банківські реквізити, поле для коментарію, іконка (для візуалізації замовлень та маршрутів), а також сформоване експертами нечітке число, що характеризує типову поведінку конкретного замовника (відхилення від чіткого обсягу замовленого палива у відносних одиницях). В класі *TCustomer* перевизначаються всі успадковані віртуальні процедури та конструктори, відповідно до його специфічної функціональності.

Клас "Дислокація" (*TDislocation*) також є нащадком класу *TDataObject*. Його задача полягає в збереженні географічної інформації про можливий пункт розміщення замовлення. До специфічних властивостей класу *TDislocation* відносяться: повна назва пункту дислокації, координати x , y (в пікселях) та іконка для візуалізації на мапі. При цьому всі успадковані віртуальні процедури та конструктори перевизначено.

Об'єкти класу "Замовлення" (*TDemand*) містять інформацію про конкретні замовлення. Через зв'язки *Dislocation* та *Customer* об'єкти даного класу здійснюють зв'язок з асоційованими "Замовниками" та "Дислокаціями". Потужність обох зв'язків завжди "один до одного", тобто одному "Замовленню" відповідає один "Замовник" в одній "Дислокації". Видимість пов'язаних об'єктів забезпечується через одноіменні поля типу *TKeyId*. Поле *CrispDemand* дійсного типу визначає обсяг чіткого замовлення, а поле *FuzzyDemand* – обсяг нечіткого замовлення.

Для групування об'єктів даних розроблено відповідні класи (бази даних), що представлені на рис. 3.2.

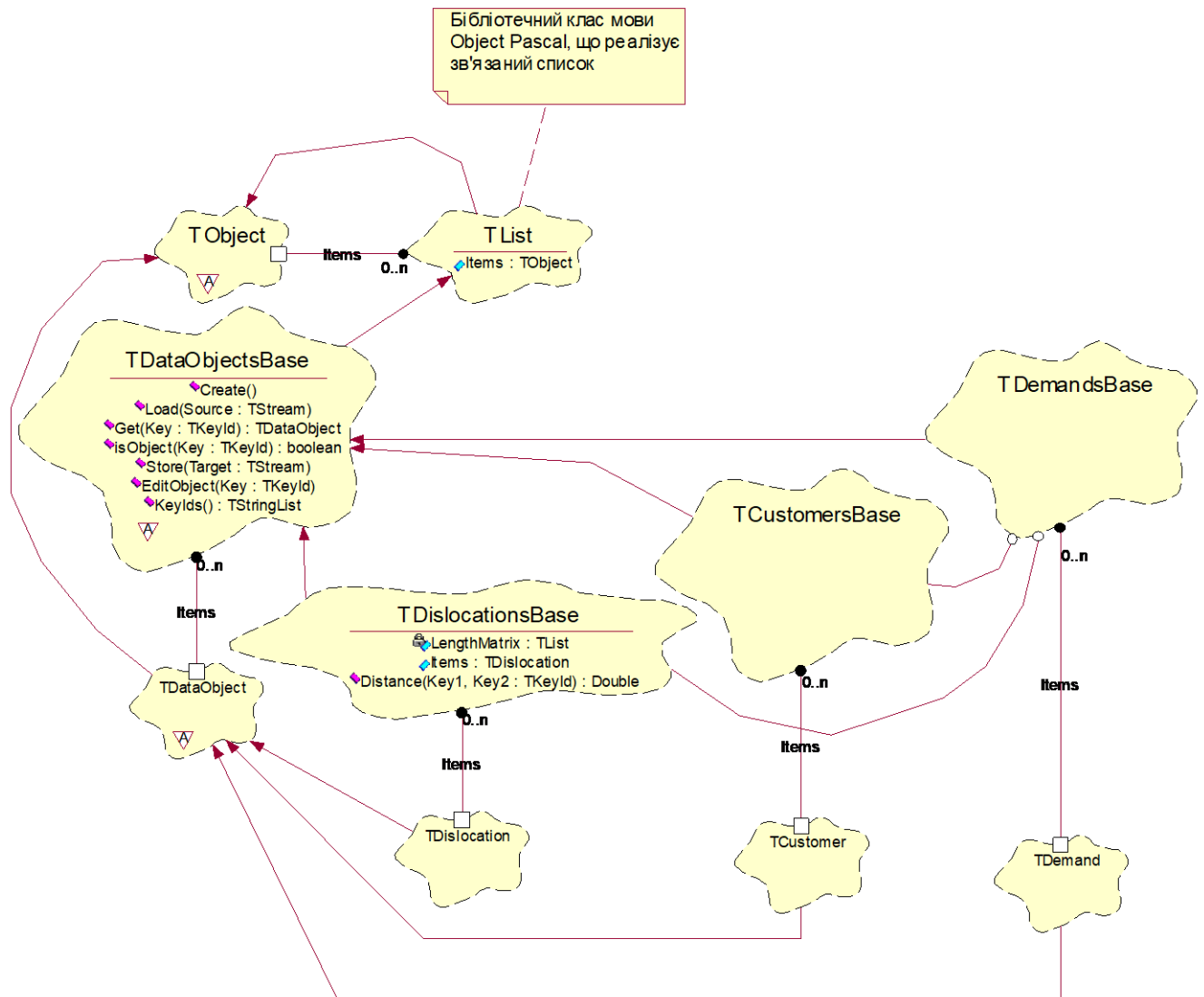


Рис. 3.2. Діаграма класів модуля *DataBases*

До складу модуля *DataBases* входять: база замовників (*TCustomerBase*), База дислокацій (*TDislocationBase*) та База замовлень (*TDemandBase*). Так як функціональність всіх трьох баз в основному співпадає, то для уніфікації звертань до класів-баз введено спільний предок – клас *TDataObjectBase*, що відтворює загальні функціональні операції баз, оперуючи при цьому з абстрактним класом *TDataObject*. Семантика зв'язку між цими двома класами визначається як агрегація ("ціле – частина") з потужністю "один до багатьох", тобто одна база може містити багато об'єктів даних.

Перелік операцій абстрактної бази *TDataObjectBase* визначено як:

- конструктор *Create*, що створює нову порожню базу;
- абстрактний конструктор *Load*, що призначений для завантаження бази з потоку. В базовому класі даний конструктор завантажує з потоку параметри бази, що успадковуються від класу *TList*. Зчитування об'єктів – елементів бази покладається на конструктор *Load* класів-нащадків, так як в класі *TDataObjectBase* безпосередньо визначити клас об'єктів даних неможливо;
- функція *Get(Key: TKeyId)*, що повертає фізичну адресу (вказник) об'єкта даних за його унікальним ідентифікатором (якщо об'єкт відсутній, повертається значення **nil**) ;
- функція *isObject(Key: TKeyId)*, що повертає булівське значення "Істина", якщо в базі існує об'єкт з заданим значенням унікального ідентифікатора *Key*;
- функція *KeyIds()*, що повертає список (бібліотечний клас *TStringList*) унікальних ідентифікаторів всіх існуючих в базі об'єктів даних;
- віртуальна процедура *Store*, що призначена для збереження бази в потоці. В базовому класі задача цієї процедури полягає в виведенні до потоку параметрів бази, що успадковуються від класу *TList*, та розсиланні відповідного повідомлення всім об'єктам, що входять до складу бази;
- віртуальна абстрактна процедура *EditObject*, що викликає форму для редагування об'єкта даних. Параметром цієї процедури є значення унікального ідентифікатора.

Спеціалізовані бази (*TCustomerBase*, *TDislocationBase* та *TDemandBase*) перевизначають частину операцій базового класу, відповідно до їх функціональності. Крім того, клас *TDislocationBase* додатково містить операцію *Distance* для визначення відстані між двома дислокаціями, заданими своїми унікальними ідентифікаторами. В якості результату дана функція повертає одержане з матриці відстаней дійсне число. З метою підвищення стабільності роботи модуля відповідне поле *LengthMatrix* віднесено до приватних елементів класу *TDislocationBase*. Для кожної бази додатково розроблено класи-форми редагування об'єктів даних. Зовнішній вигляд однієї з вказаних форм, а саме форми для редагування замовлення, представлено на рис. 3.3.

The image shows a software dialog box titled "Замовлення №3". It has a standard Windows-style title bar with minimize, maximize, and close buttons. The dialog contains several input fields and a graph. The "Замовник" field is a dropdown menu with "АО 'Укррічфлот'" selected. The "Дислокація" field is another dropdown menu with "Порт" selected. Below these is a text field for "Обсяг замовлення:" containing the value "368.00". A central graph area features a blue dashed grid with a red triangle overlaid on it. The triangle's base is on the x-axis, and its peak is at the top of the grid. Below the graph are three input fields labeled "Мінімальне", "Середнє", and "Максимальне", with values "212.00", "368.00", and "450.00" respectively. At the bottom of the dialog are three buttons: "ОК", "Застосувати", and "Скасувати".

Рис. 3.3. Діалогове вікно – форма для редагування замовлень

Як видно з рис. 3.3, дизайн діалогових вікон повністю відповідає сучасним вимогам щодо ергономіки та діючих стандартів.

На рис. 3.4. представлена діаграма класів модуля *FuzzyClasses*, що призначений для реалізації НЕС. З боку інших модулів системи планування маршрутів зв'язки існують лише з класами *TFuzzySystem* та *TMF*, тому в інтерфейсну частину модуля винесено лише ці класи. В таблиці 3.1. наведено функціональність властивостей і методів класу *TFuzzySystem*, що реалізує нечітку експертну систему.

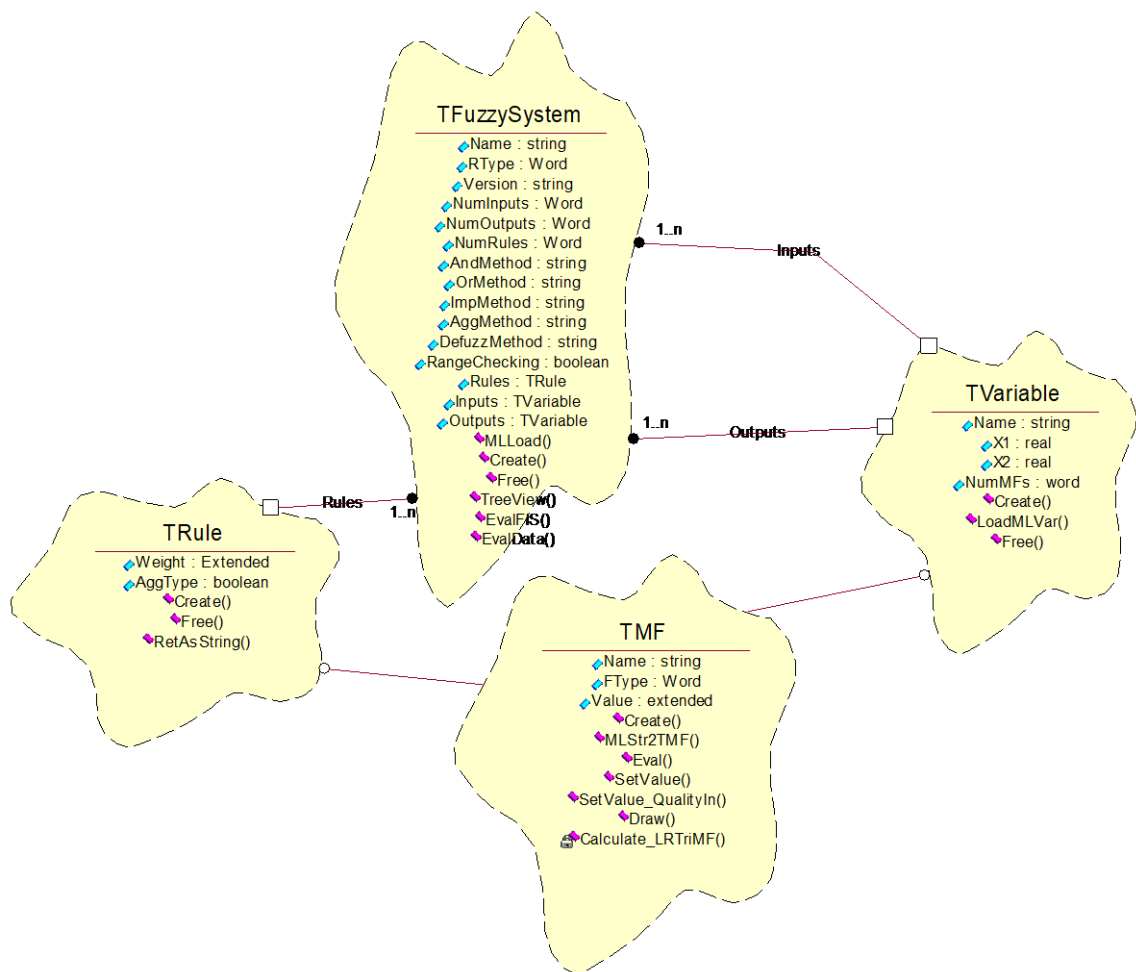


Рис. 3.4. Діаграма класів модуля *FuzzyClasses*

Таблиця 3.1

Властивості та методи класу *TFuzzySystem*

№	Ідентифікатор	Тип	Функціональність
1.	<i>Name</i>	<i>string</i>	Ідентифікатор нечіткої експертної системи.

2.	<i>RType</i>	<i>word</i>	Тип алгоритму нечіткого логічного виводу, може приймати значення <i>rtSugeno</i> або <i>rtMamdani</i>
3.	<i>Version</i>	<i>string</i>	Версія системи нечіткого логічного виводу (введено для сумісності з середовищем MatLab)
4.	<i>NumInputs</i>	<i>word</i>	Число входів нечіткої експертної системи
5.	<i>NumRules</i>	<i>word</i>	Число правил нечіткої експертної системи
6.	<i>AndMethod</i>	<i>string</i>	Метод обчислення нечіткої логічної операції "І", може приймати значення: ' <i>Zadeh</i> ', ' <i>Werners</i> ', ' <i>ZimmZisno</i> '.
7.	<i>OrMethod</i>	<i>string</i>	Метод обчислення нечіткої логічної операції "АБО", може приймати значення: ' <i>Zadeh</i> ', ' <i>Werners</i> ', ' <i>ZimmZisno</i> '.
8.	<i>AggMethod</i>	<i>string</i>	Метод агрегації правил, може приймати значення ' <i>and</i> ' або ' <i>or</i> '
9.	<i>DefuzzMethod</i>	<i>string</i>	Метод дефазифікації – може приймати одне з значень: ' <i>Centroid</i> ' – метод центра ваги, ' <i>LeftMax</i> ' – метод найлівішого максимуму, ' <i>RightMax</i> ' – метод найправішого максимуму, ' <i>BySector</i> ' – метод медіан
10.	<i>Rules</i>	<i>TRule</i>	База лінгвістичних правил
11.	<i>MLLoad</i>	<i>операція</i>	Конструктор, що дозволяє завантажувати НЕС з стандартних файлів середовища MatLab
12.	<i>Free</i>	<i>операція</i>	Деструктор
13.	<i>TreeView</i>	<i>операція</i>	Операція візуалізації НЕС за допомогою стандартного діалогового елемента <i>TTreeView</i>
14.	<i>EvalFIS</i>	<i>операція</i>	Операція обчислення значення вихідного сигналу нечіткої експертної системи для одного вектора вхідних даних
15.	<i>EvalData</i>	<i>операція</i>	Високопродуктивна операція швидкісної обробки великих масивів вхідних даних

Структура модуля *MainProgram* наведена на рис. 3.5. До цього модуля входять основні класи інтелектуальної інформаційної системи, зокрема *TCVRPSystem* – головний клас програми, *TBoukneringProgram* – бункерувальна програма, *TRoutes* – план маршрутів, *TRoute* – маршрут та *TSimulator* – клас, що розв’язує задачу імітаційного моделювання запланованих маршрутів, включаючи формування стохастичних моделей замовлень.

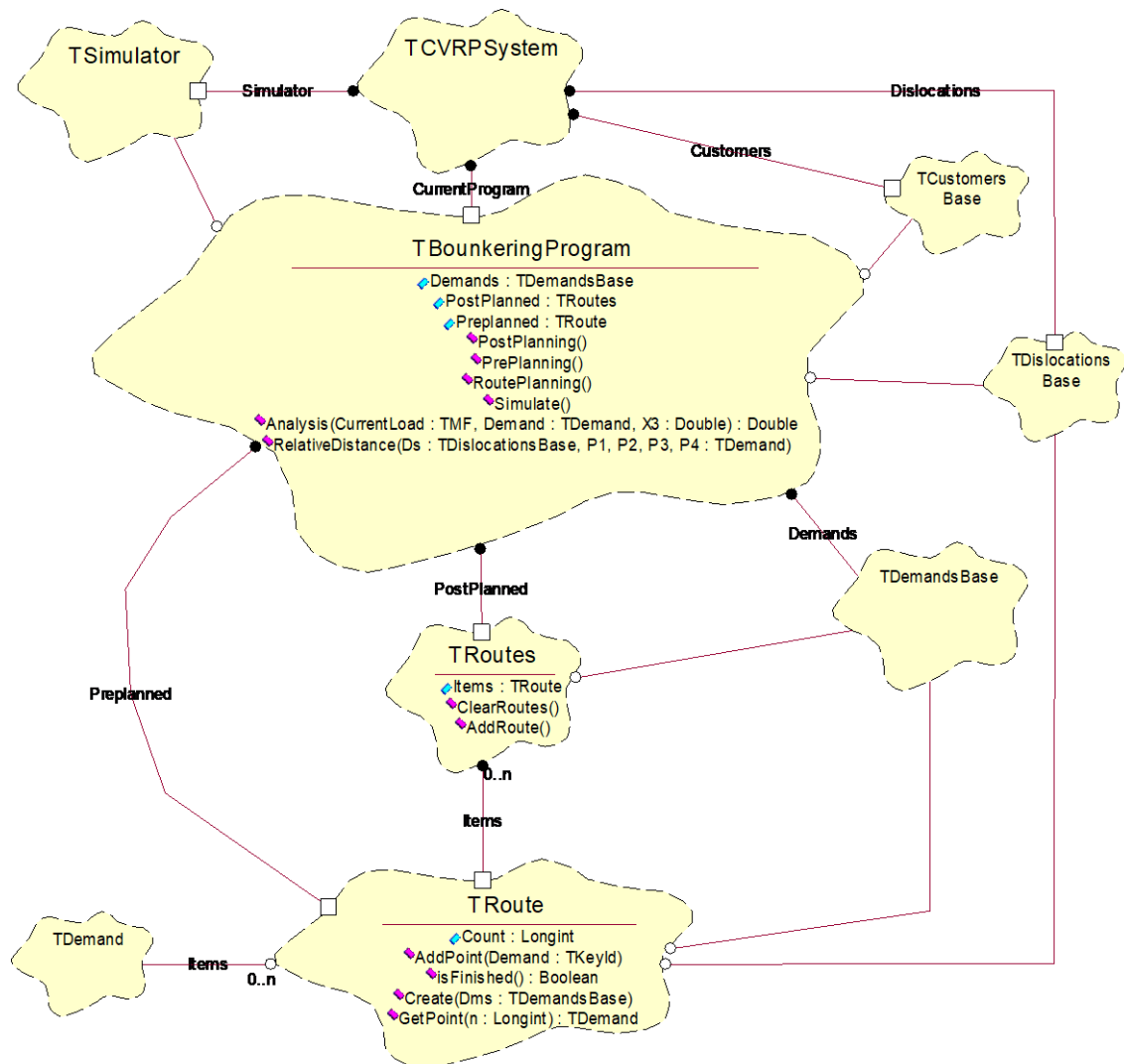


Рис. 3.5. Діаграма класів модуля *MainProgram*

Клас *TRoute* інтерпретує маршрут як список унікальних ідентифікаторів замовлень. Даний клас дає змогу відстежувати виконання замовлень (функція *isFinished* повертає булівське значення "Істина", якщо обслуговано всі замовлення маршруту) та знаходити наступне від визначеного необслуговане

замовлення (функція *NextFree*, аргументом якої є унікальний ідентифікатор поточного завантаження, функція повертає вказник на необслуговане замовлення або значення "**nil**", якщо всі замовлення обслуговано). Функція *GetPoint(n)* повертає вказник на *n*-й пункт маршруту.

До специфічних операцій класу *TRoutes* відноситься очистка плану маршрутів (*ClearRoutes*) та додавання порожнього маршруту (*AddRoute*).

Клас *TCVRPSystem* здійснює завантаження баз замовників та дислокацій, а також забезпечує управління життєвим циклом бункерувальних програм (їх створенням, збереженням та поновленням роботи).

В наступному розділі зупинимося більш детально на специфікації та функціональності основного класу *TBounkeringProgram*.

3.3 Опис об'єктної моделі

Для вирішення специфічних задач клас *TBounkeringProgram* включає до свого складу базу замовлень (зв'язок *Demands* на діаграмі, рис. 3.5) та має доступ до баз дислокацій та замовників (зв'язки *Customers* та *Dislocations*). Видимість в останніх двох зв'язках забезпечується через клас *TCVRPSystem*, що передає посилання на відповідні об'єкти при створенні екземпляру класу *TBounkeringProgram*.

Поле *Preplanned* зберігає базовий маршрут, сформований шляхом розв'язання класичної задачі комівояжера TSP (без урахування максимального завантаження танкера) для поточного списку замовлень за допомогою процедури *Preplanning*. В даній версії програмного продукту реалізовані алгоритми розв'язку задачі комівояжера шляхом моделювання випалювання та за економічним Clark і Wright алгоритмом.

Поле *Postplanned* містить оптимальний план виконання замовлень з урахуванням максимального завантаження транспортної одиниці і формується в результаті виконання процедури *PostPlanning*.

Алгоритм процесу планування маршрутів подано на рис. 3.6.

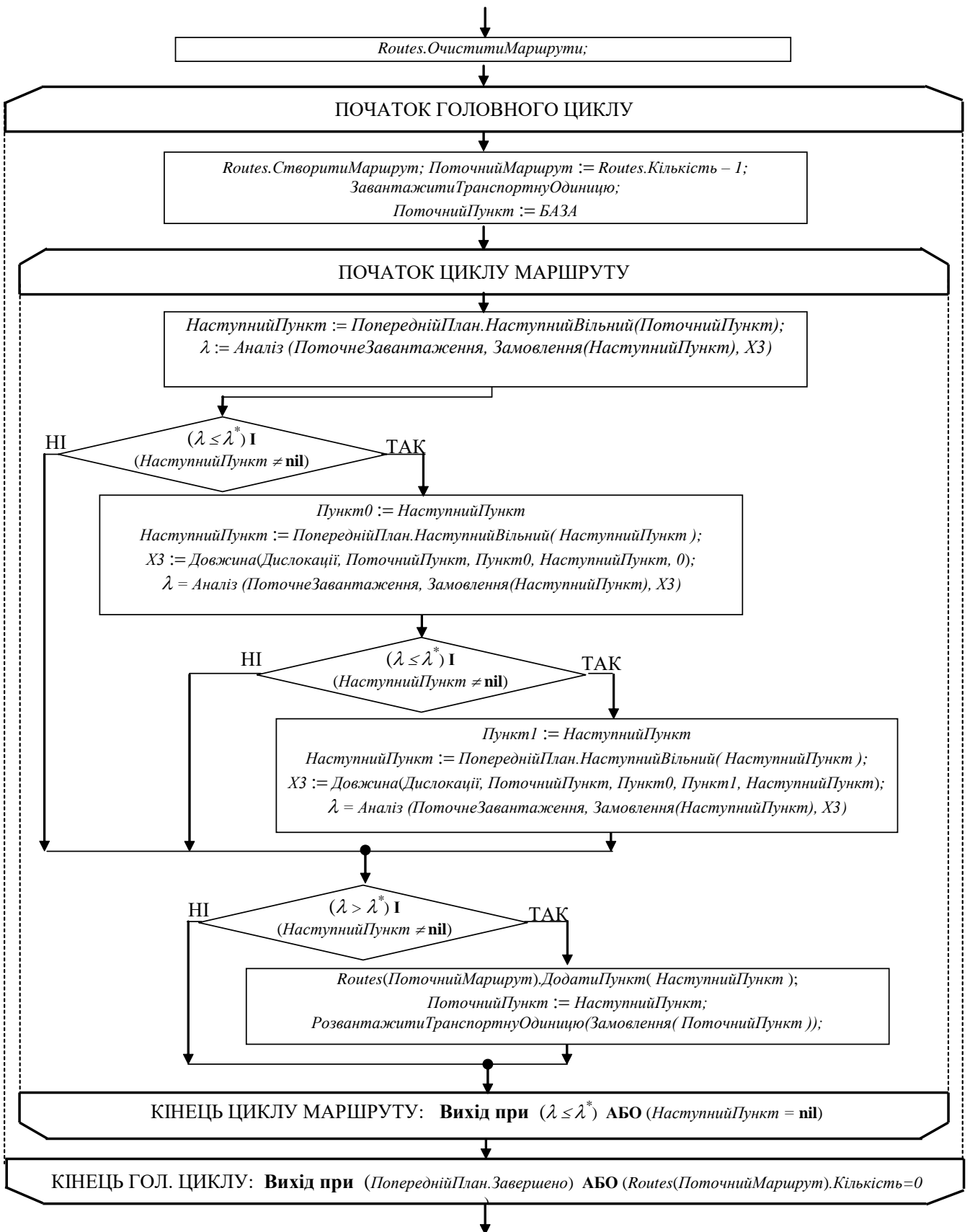


Рис. 3.6. Алгоритмічна схема процесу планування логістичних маршрутів з урахуванням завантаження

В приведеній схемі використовуються функції *Analysis* та *RelativeDistance*, що також відносяться до операцій класу *TBunkeringProgram*.

Функція *Analysis* призначена для обчислення поточного значення рівня сатисфакції λ і використовується при перевірці можливості включення вузла до поточного маршруту. Її вхідними параметрами є нечіткі числа, що відповідають поточному завантаженню танкера та обсягу замовлення в наступному вузлі, а також значення відносної довжини x_3 для альтернативних варіантів рішень.

При обчисленні рівня сатисфакції здійснюється перевірка на наявність конфліктної ситуації. Якщо конфліктна ситуація відсутня, то значення λ встановлюється в 0 чи 1 без виклику нечіткої експертної системи. В іншому випадку викликається нечітка експертна система, вхідними параметрами якої виступають нечіткі значення поточного завантаження танкера, обсяг замовлення в наступному пункті а також максимальне завантаження танкера і відносна довжина x_3 .

Алгоритм аналітичної фазифікації *LR*-числа (результату ділення нечітких чисел з трикутною ФН) лінгвістичним термом трикутної форми реалізовано в класі *TFuzzySystem*.

Значення відносної довжини x_3 обчислюється функцією *RelativeDistance*, аргументами якої виступають вказники на замовлення в 4-х вузлах маршруту та база дислокацій. Ідентифікатор дислокації базового порту при обчисленні функції *RelativeDistance* визначається за значенням поля *Base*, що є загальним для всієї бункерувальної програми.

Процедури *RoutePlanning* та *Simulate* представляють собою сценарії для процесів планування маршрутів та імітаційного моделювання їх реалізації.

Аналогічні підходи застосовано при розробці повної об'єктної моделі інтелектуальної системи планування маршрутів і траєкторій, програмну реалізацію якої на основі мови Object Pascal наведено в додатку.

Висновки до розділу

В даному розділі на основі об'єктно-орієнтованого підходу розроблено прототип СППР у вигляді інформаційної системи для планування логістичних маршрутів на основі об'єктно-орієнтованої технології з реалізацією об'єктних моделей: нечіткої експертної системи, класичної задачі комівояжера, формування оптимальних маршрутів з урахуванням невизначеності замовлень (задача CVRP). Сформовано об'єктні моделі представлення даних, зокрема, бази замовлень, бази замовників, бази можливих дислокацій замовників.

ВИСНОВКИ

В магістерській роботі виконано дослідження моделей, методів та інформаційної системи побудови логістичних маршрутів на основі нечітких моделей. В роботі показано, що підвищення ефективності процесів формування та оптимізації логістичних маршрутів і траєкторій в умовах невизначеності можливе шляхом удосконалення існуючих моделей і алгоритмів для прийняття рішень в конфліктних ситуаціях при різних ступенях невизначеності поточної інформації.

Описані моделі, алгоритми та програмне забезпечення забезпечують підвищення ефективності використання обчислювальних ресурсів, підвищення швидкодії алгоритмів, розробки нових та удосконалення існуючих методів формалізації, оцінки ефективності і моделювання синтезованих в умовах невизначеності маршрутів і траєкторій.

На основі застосування нечіткої логіки розроблено узагальнену структуру багатокрокових алгоритмів планування та оптимізації маршрутів і траєкторій з використанням процедур ранжування вузлів.

Розроблено структуру інтелектуальної інформаційної системи, базу знань НЕС для прийняття рішень шляхом прогнозування рівня сатисфакції нечітких замовлень в вузлах-претендентах та процедуру вибору критичного значення рівня сатисфакції на основі мінімізації загальної довжини маршрутів.

Описано узагальнений алгоритм фазифікації вхідної інформації при застосуванні нечітких моделей вхідних сигналів та лінгвістичних термів у вигляді нечітких множин.

Розроблено прототип інтелектуальної системи для планування логістичних маршрутів на основі об'єктно-орієнтованої технології з реалізацією об'єктних моделей: нечіткої експертної системи, класичної задачі комівояжера, формування оптимальних маршрутів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Зайченко Ю.П. Дослідження операцій. – Київ: ЗАТ “Віпол”, 2000. – 688 с.
2. Кутковецький В.Я. Дослідження операцій. – Миколаїв, МФ НаУКМА, 2002. – 182 с.
3. Пічугін Є.Д. Методи оптимізації. – Одеса, ОГПУ, 1998. – 196 с.
4. Таха Х. Вступ в дослідження операцій. Кн. 1. Пер. з англ. – К.: 1999. – 479 с.
5. Bodin, L., Golden, B., Assad, A. and Ball, M. Routing and Scheduling of Vehicles and Crews: The State of the Arts, An Intern. Journal Computers & Operations Research, Vol.10, No.2, 1983. – pp. 79-115.
6. Laporte G. The Traveling Salesman Problem: An overview of exact and approximate algorithms. Europ. Journ. of Operational Research 59 ,1992. - pp. 231-248.
7. Laporte G. The Vehicle Routing Problem: An overview of exact and approximate algorithms. European Journal of Operational Research 59 ,1992. – pp. 345-348.
8. Teodorovic D., Pavkovich G. The fuzzy set theory approach to the vehicle routing problem when demand at nodes is uncertain. Fuzzy Sets and Systems, 82, 1996. – PP. 307 – 317.
9. Tan, B. and Otay Emra N. Modelling and analysis of vessel casualties resulting from the tanker traffic through narrow waterway. Naval Research Logistics, Vol. 45, No. 8, 1999, pp.871-892.
10. Christer, A.H. and Lee, S.K. Modelling ship operation reliability over a mission under regular inspections. Journal of Operation Research Society, Vol. 48, No. 7, 1997, pp. 688-699.
11. Alanc, L.A. and Rucks, C.T. (1996), A multy discriminant analysis of vessel accidents. Accid. Anal. and Prev. Vol. 28, No. 4, pp. 501-510.
12. Ketkar, K.W. and Babu, A.J.G. (1997), An analysis of oil spills from vessel traffic accidents. Transportation Research D, Vol. 2, No. 1, pp. 35-41.

13. Rana, K. and Vickson, R.G. A Model and Solution Algorithm for Optimal Routing of Time-Chartered Containership, *Transportation Science*, Vol.22, No.2, 1988, pp. 83-95.
14. Hanif D., Salem M.A.-Y. and Merza M.H. Fleet management models and algorithms for an oil-tanker routing and scheduling problem. *IIE Transactions*, Vol. 31, 1999, pp.395-406.
15. Michalewicz Z. Fogel D.B. *How to solve it: modern heuristic*, Berlin, Springer, 1999.
16. Глибовець М.М., Олецкий О.В. *Штучний інтелект*. – К.: Видавничий дім «КМ Академія», 2002. – 369 с.
17. Jeromin B., Korner F. Triangle inequality and symmetry in connection with the assignment and the traveling salesman problem. *European Journal of Operational Research*, 38, 1989. – pp.70-75.
18. Gendreau M., Hertz A., Laporte G. New insertion and postoptimization procedure for the traveling salesman problem. *Operations Research*, Vol. 40, No. 6, 1992. – pp.1086-1094.
19. Herer Y.T. Submodularity and the traveling salesman problem. *European Journal of Operational Research*, 114, 1999. – pp.489-508.
20. Glover F., Greenberg H.J. New approaches for heuristic search: A bilateral linkage with artificial intelligence. *European Journal of Operational Research*, 39, 1989. – pp.119-130.
21. Michalewicz, Z. *Genetic algorithms + data structures = evolution programs*, Berlin, Springer, 2000.
22. Laporte G., Mercure H., Nobert Y. A branch-and-bound algorithms for a class of asymmetrical capacitated vehicle routing problems. *Journal of the Operational Research Society*, 1991.
23. Christofides N., Mingozzi A., Toth P. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, No. 20, 1981. – pp.255-282.

24. Junginger W. On representative of multi-index transportation problems. *European Journal of Operational Research*, 66, 1993. – pp.353-371.
25. Gillett B.E., Miller L.R. A Heuristic Algorithm for the Vehicle Routing Problem. *Operations Research*, No. 22, 1974. – pp. 340-349.
26. Laporte G., Mercure H., Nobert Y. An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks*, 16, 1986. –pp. 33-46.
27. Змитрович А.І. Інтелектуальні інформаційні системи. Мінськ: Тетра Системс, 1997. – 368 с.
28. Gendrau M., Laporte G., Seguin R. A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers. *Operations Research*, Vol. 44, No. 3, 1996. – pp. 469-477.
29. Taillard E.D., Laporte G., Gendrau M. Vehicle Routeing with Multiple Use of Vehicles. *Journal of the Operational Research Society*, No. 47, 1996. – pp. 1065-1070.
30. Рідкокаша А.А., Голдер К.К. Основи систем штучного інтелекту. Черкаси: Відлуння – Плюс, 2001. – 240 с.
31. Хант І. Штучний інтелект. Пер. з англ. К., 2008. – 560 с.
32. Архангельский В.І., Богаенко І.Н., Грабовский Г.Г., Рюмшин Н.А. Нейронні мережі в системах автоматизації. – Київ, Техніка, 1999. – 364 с
33. Zadeh L. Fuzzy Sets. *Information And Control*. –1965, No. 8. – PP. 338-353.
34. Zimmerman H.J. *Fuzzy Set Theory and Its Applications*. – Kluwer Academic Publishers, Boston/Dordrecht/London, 1992. – 399 p.
35. Богаенко І.М., Григорків В.С., Бойчук М.В., Рюмшин М.О. Математичне програмування. – К.: Логос, 1996. – 266 с.
36. Малахов В.П., Імітаційні моделі та алгоритми формування маршрутів і траєкторій на основі нечіткої логіки. – Праці Одеського політехнічного університету, 2002. Вип. №2(18). – сс. 72-80.
37. Пушкар О.І. Системи підтримки рішень слабоформалізованих задач розвитку підприємств. – Харків: ХДЕУ, 1997. – 140 с.

38. Jantsen J. Design of Fuzzy Controllers. Tech. Report no. 98 E 864, Technical University of Denmark, Dept. Of Automation, Lyngby, 1998. – pp. 1-26.
39. Szmidt E., Kacprzyk J. Distances between intuitionistic fuzzy sets // Int. Jo. Fuzzy Sets and Systems. --№ 114, 2000. – pp.505-518.
40. Zimmermann H.-J. , Zysno P. Latent connectives in human decision making. Fuzzy Sets and Systems 3, 1979. - pp. 37-51.
41. Negoita C.V., Sularia M. On fuzzy mathematical Programming and tolerances in planning, Economic Computation and Economic Cybernetics Studies and Research 3, 1976. – pp. 3-15.
42. Desrosiers J., Soumis F., Desrosiers M., Sauve M. Methods for routing with time windows. European Journal of Operational Research 23, 1986. - pp. 236-245.
43. Стогній Б.С., Кириленко О.В., Липківський К.О., Кирик В.В. Fuzzy-системи – новий технологічний інструмент систем управління. Технічна електродинаміка, № 3, 2001. – с.17-20.
44. Липківський К.О., Кирик В.В. Основні підходи до створення fuzzy-систем управління регуляторами змінної напруги. Праці ІЕД НАН України, 2001. – с.13-16.
45. Kolen A.W.J., Rinnooy Kan A.H.G., Trienekens H.W.J.M. Vehicle routing with time windows. Operations Research, Vol. 35, No. 2, 1987. – pp. 266-273.

ДОДАТКИ

Додаток А.

Фрагмент коду модуля *FuzzyClasses*

```

unit FuzzyMF;
interface
uses classes, FuzzyCons, FuzzyMisc, extctrls, windows, graphics;
type TParams = array [1..5] of extended;
    PParams = ^TParams;
type TMF = class (TObject)
    Name: string;
    FType: Word;
    Params: TParams;
    Value: extended;
    constructor Create(NN : string; FT: Word; np: TParams);
    constructor MLStr2TMF(MLStr: string; var ErrFlag: word);
    procedure Eval(X: extended; var y: extended);
    procedure SetValue(X: extended);
    procedure SetValue_QualityIn(X: pointer; var UsedNumbers: Byte);
    procedure Draw(Canvas: TCanvas);
private
    function Calculate_LRTrimF(A,B,C: TParams): Extended;
end;

var NilMF: TMF;
implementation
{ TMF ----- Membership Function}
procedure TMF.SetValue(X: extended);
begin
    Eval(X, Value);
end;

function TMF.Calculate_LRTrimF(A,B,C: TParams): Extended;
var
    Tau: array [1..8] of extended;
    Crossings: array [1..8] of extended;
    i: integer;
    max: Extended;
begin
    Tau[1] := (sqr(B[3])*sqr(C[2])-2*B[3]*C[2]*C[1]*B[2]-
        2*B[3]*C[2]*A[2]-2*B[3]*C[2]*A[1]+sqr(C[1])*sqr(B[2])-
        2*C[1]*B[2]*A[2]-2*C[1]*B[2]*A[1]+sqr(A[2])-
        2*A[2]*A[1]+sqr(A[1])+4*B[2]*A[1]*C[2]+4*B[3]*C[1]*A[2]);
    Tau[2] := Tau[1];
    Tau[3] := (-2*B[3]*C[2]*A[2]-2*B[3]*C[2]*A[1]+sqr(B[3])*sqr(C[2])+
        sqr(A[2])-2*A[2]*A[1]+sqr(A[1])+4*A[2]*C[3]*B[3]-
        2*C[3]*B[2]*A[1]-2*A[2]*C[3]*B[2]+sqr(C[3])*sqr(B[2])-
        2*B[3]*C[2]*C[3]*B[2]+4*B[2]*A[1]*C[2]);
    Tau[4] := Tau[3];
    Tau[5] := (-2*C[3]*B[2]*A[3]-2*C[2]*B[1]*A[3]+4*A[2]*C[3]*B[1]-
        2*A[2]*C[2]*B[1]+sqr(A[2])-2*A[2]*A[3]+sqr(A[3])-
        2*C[2]*B[1]*C[3]*B[2]+sqr(C[2])*sqr(B[1])-
        2*A[2]*C[3]*B[2]+sqr(C[3])*sqr(B[2])+4*C[2]*B[2]*A[3]);
    Tau[6] := (-2*C[3]*B[2]*A[3]-2*C[2]*B[1]*A[3]+4*A[2]*C[3]*B[1]-
        2*A[2]*C[2]*B[1]+sqr(A[2])-2*A[2]*A[3]+sqr(A[3])-
        2*C[2]*B[1]*C[3]*B[2]+sqr(C[2])*sqr(B[1])-
        2*A[2]*C[3]*B[2]+sqr(C[3])*sqr(B[2])+4*C[2]*B[2]*A[3]);
    Tau[7] := (-2*C[2]*B[1]*A[3]-2*C[1]*B[2]*A[2]-2*A[2]*C[2]*B[1]-
        2*C[2]*B[1]*C[1]*B[2]-2*C[1]*B[2]*A[3]+
        sqr(C[1])*sqr(B[2])+sqr(A[2])-2*A[2]*A[3]+sqr(A[3])+
        sqr(C[2])*sqr(B[1])+4*A[2]*C[1]*B[1]+4*C[2]*B[2]*A[3]);

```

```

Tau[8]:= (-2*C[2]*B[1]*A[3]-2*C[1]*B[2]*A[2]-2*A[2]*C[2]*B[1]-
2*C[2]*B[1]*C[1]*B[2]-2*C[1]*B[2]*A[3]+
sqr(C[1])*sqr(B[2])+sqr(A[2])-2*A[2]*A[3]+sqr(A[3])+
sqr(C[2])*sqr(B[1])+4*A[2]*C[1]*B[1]+4*C[2]*B[2]*A[3]);

for i:=1 to 8 do if Tau[i]<0 then Tau[i]:=0;

Crossings[1] := 1/2/(C[2]*B[2]+C[1]*B[3]-B[3]*C[2]-C[1]*B[2])*
(A[2]-A[1]+2*C[1]*B[3]-B[3]*C[2]-C[1]*B[2]+ Sqrt(Tau[1]));
Crossings[2] := 1/2/(C[2]*B[2]+C[1]*B[3]-B[3]*C[2]-C[1]*B[2]) *
(A[2]-A[1]+2*C[1]*B[3]-B[3]*C[2]-C[1]*B[2]- Sqrt(Tau[2]) );
Crossings[3] := 1/2/(C[2]*B[2]+C[3]*B[3]-B[3]*C[2]-C[3]*B[2]) *
(A[2]-A[1]+2*C[3]*B[3]-B[3]*C[2]-C[3]*B[2]+ Sqrt(Tau[3]) );
Crossings[4] := 1/2/(C[2]*B[2]+C[3]*B[3]-B[3]*C[2]-C[3]*B[2]) *
(A[2]-A[1]+2*C[3]*B[3]-B[3]*C[2]-C[3]*B[2]- Sqrt(Tau[4]) );
Crossings[5] := 1/2/(C[2]*B[2]+C[3]*B[1]-C[2]*B[1]-C[3]*B[2]) *
(A[2]-A[3]+2*C[3]*B[1]-C[2]*B[1]-C[3]*B[2]+ Sqrt(Tau[5]) );
Crossings[6] := 1/2/(C[2]*B[2]+C[3]*B[1]-C[2]*B[1]-C[3]*B[2]) *
(A[2]-A[3]+2*C[3]*B[1]-C[2]*B[1]-C[3]*B[2]- Sqrt(Tau[6]) );
Crossings[7] := 1/2/(C[2]*B[2]+C[1]*B[1]-C[2]*B[1]-C[1]*B[2]) *
(A[2]-A[3]+2*C[1]*B[1]-C[2]*B[1]-C[1]*B[2]+ Sqrt(Tau[7]) );
Crossings[8] := 1/2/(C[2]*B[2]+C[1]*B[1]-C[2]*B[1]-C[1]*B[2]) *
(A[2]-A[3]+2*C[1]*B[1]-C[2]*B[1]-C[1]*B[2]- Sqrt(Tau[8]) );

for i:=1 to 8 do
    if Tau[i]<0      then Crossings[i]:=0
                    else if (Crossings[i]<0) or (Crossings[i]>1)
                        then Crossings[i]:=0;

max:=Crossings[1];
for i:=2 to 8 do if max<Crossings[i] then max:=Crossings[i];
Calculate_LRTriMF:=max;
end;

procedure TMF.SetValue_QualityIn(X: Pointer; var UsedNumbers: Byte);
type TDoubleParams = array [1..10] of double;
    PDoubleParams = ^TDoubleParams;

var XCrisp: Double;
    XFuzzy: PDoubleParams;

var a1,a2,a3, m: Extended;
    b1,b2,b3: Double;
    A, B: TParams;
    i: word;
begin
    XCrisp := PDouble(X)^;
    XFuzzy:=X;
    if FType<First_Quality_MF
        then begin
            SetValue(XCrisp); {Quantitative input}
            UsedNumbers:=1;
            end
        else begin {Quality input}
            case FType of
                shTriTri: begin

                    a1:=Params[1]; a2:=Params[2];
                    a3:=Params[3];
                    b1:=XFuzzy^ [1]; b2:=XFuzzy^ [2];
                    b3:=XFuzzy^ [3];
                    UsedNumbers:=3;

                    if (a1>b3) then m:=0

```

```

else if (b2>a2) and (a3>=b1)
  then m:=(a3-b1)/(b2-b1+a3-a2)
  else if a2 = b2 then m:=1
  else if (b2<a2) and (b3>=a1)
    then m:=(b3-a1)/(a2-a1+b3-b2)
    else if a3<b1 then m:=0;
Value:=m;
end;
shLRTri: begin
  for i:=1 to 3 do A[i]:=XFuzzy^[i];
  for i:=4 to 6 do B[i-3]:=XFuzzy^[i];
  UsedNumbers:=6;
  Value:=Calculate_LRTriMF(A,B,Params);
end;
else begin
  Value:=0;
  UsedNumbers:=0;
end;
end;
end;

end;

procedure TMF.Draw(Canvas: TCanvas);
var Back: TRect;
var XrMin, XrMax: Real;
const YrMin: Real = -0.1;
      YrMax: Real = 1.05;

procedure real2screen(Xr, Yr: Real; var Xs, Ys: integer);
var mx, my: real;
begin
  mx:=(Back.Right - 5)/(XrMax - XrMin);
  my:=Back.Bottom/(YrMax - YrMin);
  Xs := 0 + round((Xr-XrMin)*mx);
  Ys := Back.Bottom - round((Yr-YrMin)*my);
end;

var X1, Y1, X2, Y2, i: Integer;
begin
  {Back.Left:=0;
  Back.Right:=Canvas.Width;
  Back.Top:=0;
  Back.Bottom:=Canvas.Height;}

  Back:=Canvas.ClipRect;

  XrMin:=-100;
  XrMax:=1500;

with Canvas do
begin
  Brush.Color:=clWhite;
  FillRect(Back);
  {Frame}
  Pen.Color:=clBlack;
  Pen.Style:=psSolid;
  Pen.Width:=2;
  MoveTo(1,1);
  LineTo(1, Back.Bottom-1);
  LineTo(Back.Right-1,Back.Bottom-1);
  LineTo(Back.Right-1, 1);
  LineTo(1,1);
  {grid }

```

```

Pen.Color:=clBlue;
Pen.Style:=psDot;
Pen.Width:=1;

for i:=1 to 5 do begin
  Real2Screen(i*XrMax/5, 0, X1,Y1);
  Real2Screen(i*XrMax/5, 1, X2,Y2);
  MoveTo(X1, Y1);
  LineTo(X2,Y2);
end;

for i:=1 to 5 do begin
  Real2Screen(0, i/5, X1,Y1);
  Real2Screen(XrMax, i/5, X2,Y2);
  MoveTo(X1, Y1);
  LineTo(X2,Y2);
end;

{Axes}
Pen.Color:=clBlue;
Pen.Style:=psSolid;
Pen.Width:=1;
Real2Screen(0, YrMin, X1, Y1);
Real2Screen(0, YrMax, X2, Y2);
MoveTo(X1, Y1);
LineTo(X2, Y2);
Real2Screen(-100, 0, X1, Y1);
Real2Screen(1500, 0, X2, Y2);
MoveTo(X1, Y1);
LineTo(X2, Y2);

{Membership Function}
Pen.Color:=clRed;
Pen.Style:=psSolid;
Pen.Width:=2;

if FType = shTrimf then begin
  Real2Screen(XrMin, 0, X1, Y1);
  MoveTo(X1,Y1);
  Real2Screen(Params[1], 0, X1, Y1);
  LineTo(X1, Y1);
  Real2Screen(Params[2], 1, X1, Y1);
  LineTo(X1, Y1);
  Real2Screen(Params[3], 0, X1, Y1);
  LineTo(X1, Y1);
  Real2Screen(XrMax, 0, X1, Y1);
  LineTo(X1, Y1);
end;
end;
end;
procedure TMF.Eval(X: extended; var y: extended);
begin
  case FType of
    shGaussMF: {Gaussian curve membership function.}
      begin
        y:= EXP(-sqr(X - params[2])/(2*sqr(params[1])));
      end;
    shTriMF: {Triangular membership function.}
      begin
        if (X<Params[1] ) or (x>Params[3] )
          then y:=0;
        if (X>=Params[1]) and (x<=Params[2])
          then y:=(x-params[1])/(params[2] - params[1]);
        if (X>=Params[2]) and (x<=Params[3])

```

```

        then y:=(x-params[3])/(params[2] - params[3]);
    end;
    shError: {Error reading of membership function}
    begin
        y:=0;
    end;
end;
end;
constructor TMF.Create(NN : string; FT: Word; np: TParams);
begin
    Inherited Create;
    Name:=NN;
    FType:=FT;
    Params:=np;
end;
constructor TMF.MLStr2TMF(MLStr: string; var ErrFlag: Word);
var a,b,c:byte;
    d: integer;
    k: char;
    S,s1:String;
    i,j:Word;
begin
    Inherited Create;
    s:=MLstr;
    {'mf1':'gausmf',[2.9 1.89]}
    b:=pos(':', s);
    c:=pos(',', s);
    Name:=copy(S,2, b-3);

    {Shape of MF}
    s1:=copy(S,b+2, c-b-3);
    FType:=shError;
    for i:=0 to NumberShapes do
    begin
        if UpString(S1) = UpString(Shapes[i])
            then begin
                FType:=i;
                break;
            end;
    end;
    end;
    if FType = shError then ErrFlag:=ErrFlag or 2;

    delete(s,1,c+1);
    i:=1;s1:='';j:=0;
    repeat
        inc(j);
        if (s[j] <> ' ') and (s[j] <> ']') then s1:=s1+s[j]
            else begin
                val(s1, params[i], d);
                s1:='';
                inc(i);
            end;
    until s[j] = ']';
end;
const p:TParams = (0,0,0,0,0);
begin
    NilMF:=TMF.Create('None', shNone, p);
end.

```